DOCUMENT RESUME

ED 282 533 IR 012 678

AUTHOR	Herrmann, Eric J.
TITLE	Computer Controlled Test Systems. Introduction. A Course Based on the IEEE 488 Bus.
INSTITUTION	Everett Community Coll., WA. Dept. of Electronic Engineering Technology.
PUB DATE	86
NOTE	212p.
PUB TYPE	Guides - Classroom Use - Materials (For Learner) (051) -- Computer Programs (101)
EDRS PRICE	MF01/PC09 Plus Postage.
DESCRIPTORS	*Computer Assisted Testing; *Computer Science Education; Computer Software; Course Descriptions; *Course Objectives; Engineering Education; *Engineering Technicians; Higher Education; Laboratory Manuals; Learning Laboratories; Man Machine Systems; *Programing; Study Guides; *Systems Development

ABSTRACT

An introductory course in computer automated tests and measurement systems based on the International Test Instrument-Computer Interface Standard, the IEEE (Institute of Electrical and Electronics Engineers)-488, is presented in this study guide. This course is designed to: (1) introduce the electronics engineering technician to the functional aspects of IEEE-488 based systems; (2) familiarize the technician with setup requirements for such systems; (3) enable the technician to understand test programs and write rudimentary test programs based on hands-on experience; and (4) develop in the technician a working background with IEEE-488 bus compatible equipment that will enable him/her to select instruments for rudimentary test systems. The material covers three main topics--hardware characteristics, software characteristics, and interface bus characteristics. Each weekly lesson for the 10-week course includes an overview and discussion of the lesson topic and laboratory exercises. Appended materials include solutions to the laboratory exercises, an 11-page reference list and bibliography, a suggested equipment list, a list of seven suggested readings from the bibliography, and both questions and solutions for a suggested final examination for the course. (Author/MES)

****
*   Reproductions supplied by EDRS are the best that can be made   *
*   from the original document.   *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

- Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED282533

COMPUTER CONTROLLED TEST SYSTEMS

INTRODUCTION

A COURSE BASED ON THE IEEE 488 BUS

BY

ERIC J. HERRMANN

DEPARTMENT OF ELECTRONIC ENGINEERING TECHNOLOGY

EVERETT COMMUNITY COLLEGE

EVERETT, WASHINGTON

1986

IR012678

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
ERIC.J.Herrmann

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

BEST COPY AVAILABLE

COMPUTER CONTROLLED TEST SYSTEMS

INTRODUCTION

A COURSE BASED ON THE IEEE 488 BUS

BY

ERIC J. HERRMANN

DEPARTMENT OF ELECTRONIC ENGINEERING TECHNOLOGY

EVERETT COMMUNITY COLLEGE

EVERETT, WASHINGTON

1986

PREFACE

THE PURPOSE OF THIS COURSE IS TO INTRODUCE THE ELECTRONIC ENGINEERING TECHNICIAN TO COMPUTER CONTROLLED TEST SYSTEMS BASED ON THE WIDLY ACCEPTED IEEE 488 INSTRUMENTATION INTERFACE STANDARD. CONCEPTS RELATED TO INSTRUMENT CONTROL UTILIZING THE IEEE 488 WILL BE PRESENTED IN A VIEW FOIL DRIVEN LECTURE FORMAT UTILIZING A STUDENT BOOKLET WHICH CONTAINS COPIES OF ALL CLASS VIEW FOILS. LABORATORY EXERCISES ARE INCLUDED TO REINFORCE THESE CONCEPTS.

BECAUSE THE IEEE 488 INSTRUMENTATION INTERFACE STANDARD DOES NOT SPECIFY OPERATIONAL STANDARDS (THESE ARE LEFT UP TO INDIVIDUAL MANUFACTURERS DISCRESSION). ONCE AN INDIVIDUAL HAS EXPERIENCE WORKING WITH ONE MAKE OF A PARTICULAR TYPE OF INSTRUMENT, OPERATION OF ANOTHER MAKE OF THAT TYPE OF INSTRUMENT ONLY REQUIRES LEARNING THE OPERATIONAL SPECIFICS OF THE NEW INSTRUMENT. WITH THIS IN MIND, THE LAB EXERCISES AND SOME OF THE LECTURE MATERIAL IS INSTRUMENT SPECIFIC. THIS WILL ALLOW THE COURSE TO ADDRESS CONCEPTS TO THE DETAIL LEVEL, WHICH WILL ALLOW THE STUDENTS TO DEVELOP A MORE COMPREHENSIVE UNDERSTANDING. THE INSTRUMENTS REFERENCED IN THIS COURSE WERE CHOSEN BY THE FOLLOWING CRITERIA: WIDE USE IN INDUSTRY, REASONABLE COST, AND VERSITILITY. DUE TO THE COSTS INVOLVED IN SETTING UP LAB STATIONS WITH THIS TYPE OF EQUIPMENT, SOME INSTITUTIONS MAY NOT BE ABLE TO AFFORD TO OFFER THE LAB PORTION OF THIS COURSE. THIS WAS TAKEN INTO ACCOUNT IN THE WRITING OF THIS CURRICULUM. SOLUTIONS TO ALL LAB EXERCISES ARE INCLUDED, SO THE COURSE MAY BE TAUGHT LECTURE ONLY.

THE COURSE IS DESIGNED TO BE TAUGHT IN ONE (10 WEEK) COLLEGE QUARTER. A TOTAL OF 20 HOURS ARE DEDICATED TO LECTURE, AND 40 HOURS OF SCHEDULED LAB. IT IS RECOMMENDED TO HAVE ADDITIONAL LAB TIME, ESPECIALLY IF ONLY A FEW LAB STATIONS ARE AVAILABLE.

THE MAJOR OBJECTIVES OF THIS COURSE ARE AS FOLLOWS:

- TO INTRODUCE THE ELECTRONICS ENGINEERING TECHNICIAN TO THE FUNCTIONAL ASPECTS OF IEEE 488 BASED TEST AND MEASUREMENT SYSTEMS.
- TO FAMILIARIZE THE ELECTRONICS ENGINEERING TECHNICIAN WITH SETUP REQUIREMENTS FOR IEEE 488 BASED SYSTEMS.
- TO ENABLE THE ELECTRONICS ENGINEERING TECHNICIAN TO UNDERSTAND TEST PROGRAMS AND WRITE RUDAMENTARY TEST PROGRAMS BASED ON HANDS-ON EXPERIENCE.
- TO DEVELOP IN THE ELECTRONICS ENGINEERING TECHNICIAN A WORKING BACKGROUND WITH IEEE 488 BUS COMPATABLE EQUIPMENT AS TO ENABLE THE TECHNICIAN TO SELECT INSTRUMENTS FOR RUDAMENTARY TEST SYSTEMS.

AT THE END OF EACH WEEKS LESSON ARE A SET OF SELF EXAMINATION QUESTIONS, WITH SOLUTIONS AT THE END OF THE STUDY GUIDE.

A STUDENT ENTERING THIS COURSE SHOULD HAVE A WORKING KNOWLEDGE OF BASIC LANGUAGE, AND A BACKGROUND IN THE FOLLOWING ELECTRONIC SUBJECTS: SEMICONDUCTOR DEVICES, AND I.C. CIRCUITS - LINEAR & DIGITAL. THIS COURSE IS DESIGNED TO COME AT THE END OF THE SECOND YEAR OF STUDY IN ELECTRONICS ENGINEERING TECHNOLOGY.

THE LABORATORY EXERCISES INVOLVE A VARIETY OF ELECTRONIC CIRCUITS. THE STUDENT SHOULD DIRECT HIS/HER ATTENTION TO HOW THE IEEE 488 SYSTEM IS USED TO SOLVE THE PROBLEM INVOLVING THE CIRCUIT, RATHER THAN THE CIRCUIT ITSELF.

THIS STUDY GUIDE IS DESIGNED SO THAT EACH PAGE OF TUTORIAL TEXT MAY BE USED AS A VIEW FOIL. THIS TECHNIQUE WILL ALLOW THE STUDENTS TO FOLLOW THE INSTRUCTOR WITHOUT THE DISTRACTION AND TIME CONSUMING EFFORT OF TAKING DETAILED NOTES. SPACE IS LEFT ON EACH PAGE FOR STUDENTS NOTES.

THE FOLLOWING SYMBOLOGY IS USED TO INDICATE THE NUMBER (N) OF THE REFERENCE (IN THE REFERENCE AND BIBLIOGRAPHY SECTION) WHERE SPECIFIC INFORMATION HAS BEEN OBTAINED: *N*

ERIC J. HERRMANN

COMPUTER CONTROLLED TEST SYSTEMS

SYLLABUS

COURSE DESCRIPTION: This is a lecture - lab course designed to provide an introduction to the concepts of computer controlled test systems, also referred to as automatic test equipment (ATE), based on the internationally accepted interface standard: IEEE 488-1978, Digital Interface for Programmable Instrumentation.

CLASS SESSION	TOPIC
1	Introduction to the IEEE 488
2	Hardware: controller characteristics
3	Hardware: instrument characteristics
4	Software: BASIC enhancements
5	Software: BASIC structured programming
6	Software: BASIC structured programming
7	Software: sample applications programs
8	IEEE 488 functional aspects
9	IEEE 488 functional aspects
10	System performance

GRADING:		
	Final exam	35%
	Lab exercises	60%
	Attendance	5%

		100%

TABLE OF CONTENTS

PAGE

TITLE PAGE i

PREFACE ii

COURSE OUTLINE (SYLLABUS) - suggested iii

TABLE OF CONTENTS iv

SECTION 1 INTRODUCTION TO THE IEEE 488 1-1

SECTION 2 HARDWARE: CONTROLLER CHARACTERISTICS 2-1

SECTION 3 HARDWARE: INSTRUMENT CHARACTERISTICS 3-1

SECTION 4 SOFTWARE: BASIC ENHANCEMENTS 4-1

SECTION 5 SOFTWARE: BASIC STRUCTURED PROGRAMMING 5-1

SECTION 6 SOFTWARE: BASIC STRUCTURED PROGRAMMING 6-1

SECTION 7 SOFTWARE: SAMPLE APPLICATIONS PROGRAMS 7-1

SECTION 8 IEEE 488 FUNCTIONAL ASPECTS 8-1

SECTION 9 IEEE 488 FUNCTIONAL ASPECTS 9-1

SECTION 10 SYSTEM PERFORMANCE 10-1

APPENDIX

A SOLUTIONS TO LAB EXERCISES LS-1

B REFERENCE AND BIBLIOGRAPHY RB-1

C SUGGESTED EQUIPMENT LIST EL-1

D SUGGESTED READING LIST RL-1

E EXAMINATION SOLUTIONS ES-1

INTRODUCTION: HISTORY

PRIOR TO IEEE 488

NO INTERFACE STANDARD EXISTED FOR CONNECTING INSTRUMENTS TOGETHER IN TEST AND MEASUREMENT SYSTEMS.

END USERS HAD TO UTILIZE CUSTOM BUILT INTERFACES TO INTERCONNECT EQUIPMENT FROM DIFFERENT OR SAME MANUFACTURERS.

INCOMPATABILITIES EXISTED IN THE FOLLOWING AREAS:
- DATA FORMAT - ASCII, BCD, BINARY, ETC.
- DATA TRANSMISSION - SERIAL, PARALLEL
- CONTROL SIGNALS - HANDSHAKING & TIMING
- SIGNAL LEVELS - LOGIC COMPATABLE, RS-232, ETC.
- MECHANICAL CONNECTION - ELECTRICAL CONNECTORS

THE MICROPROCESSOR INCREASED THE COMPLEXITY OF THE INTERFACES BY INCREASING THE PROGRAMMABILITY OF THE INSTRUMENTS.

INTERFACES WERE DEDICATED TO A SPECIFIC GROUP OF INSTRUMENTS.

COST AND COMPLEXITY INCREASED WITH THE VERSITILITY OF THE INSTRUMENTS USED IN THE SYSTEM.

NOTES:

INTRODUCTION: HISTORY

DEVELOPMENT OF THE "BUS" * *

HEWLETT PACKARD DEVELOPED A PROPOSAL FOR AN INTERFACE (1965-1974)

IEC (INTERNATIONAL ELECTROTECHNICAL COMMISSION) UTILIZED PROPOSAL FOR A BASE TO WORK FROM

IEEE (INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS) ALSO WORKED WITH THE PROPOSAL

BOTH IEC AND IEEE APPROVED BY 1975

ANSI (AMERICAN NATIONAL STANDARDS INSTITUTE) ADOPTED BY 1976

IEEE REVISED FOR CLARIFICATION 1978

IEC PUBLISHED ITS VERSION IN 1980

IEEE RECOMMENDED FORMATS IEEE-728-1982

NOTES:

INTRODUCTION: OVERVIEW OF THE STANDARD

PURPOSE

TO ALLOW INTERCONNECTION OF INSTRUMENTATION FROM DIFFERENT (OR SAME) MANUFACTURERS WITHOUT COMPATABILITY PROBLEMS OR SPECIAL ENGINEERING.

WHAT THE STANDARD DEFINES

FUNCTIONAL CHARACTERISTICS - ALL INTERFACE FUNCTIONS, SIGNAL LINES, TIMING RELATIONSHIPS FOR THE TRANSFER OF INFORMATION. THIS IS INDEPENDENT OF EQUIPMENT APPLICATION.

MECHANICAL CHARACTERISTICS - CONNECTOR TYPE, CONTACT ASSIGNMENT (PINOUT), CABLE ASSEMBLY. THIS IS INDEPENDENT OF EQUIPMENT OR APPLICATION.

ELECTRICAL CHARACTERISTICS - LOGIC LEVELS; CURRENT, TERMINATION, AND GROUNDING REQUIREMENTS. THIS IS INDEPENDENT OF EQUIPMENT OR APPLICATION.

OPERATIONAL CHARACTERISTICS - SET OF EQUIPMENT FUNCTIONS AND THEIR INTERRELATIONSHIPS. THIS IS CNOTD SPECIFIED BY THE STANDARD, AND IS DEPENDENT ON SPECIFIC EQUIPMENTS. TO HELP STANDARDIZE THIS AREA A SET OF FORMAT RECOMMENDATIONS (IEEE-728-1982) HAS BEEN DEVELOPED.

OTHER NAMES FOR IEEE 488 STANDARD

IEEE 488-1978, LATEST VERSION

ANSI MC1.1, SAME AS ABOVE

IEC 625-1, IDENTICAL TO ABOVE, EXCEPT FOR CONNECTOR

HP-IB - HEWLETT PACKARDS IMPLEMENTATION, WITH ENHANCEMENTS, TO IEEE 488

GPIB - GENERAL PURPOSE INTERFACE BUS, SAME AS IEEE 488

ASCII BUS - SAME AS IEEE 488, SO NAMED BECAUSE DATA CAN BE SENT AS ASCII CHARACTERS

HOW WELL IS THE BUS ACCEPTED?

UNIVERSAL USEAGE * * (FALL 1982)

- IEEE STANDARD IN 9 LANGUAGES
- 250 EQUIPMENT MANUFACTURES
- 14 COUNTRIES
- 2000 PRODUCTS

NOTES:

INTRODUCTION: OVERVIEW OF THE IEEE 488 STANDARD

BASIC HARDWARE CONSIDERATIONS

MAXIMUM OF 15 DEVICES MAY BE CONNECTED TO A BUS, ONE OF WHICH IS USUALLY A CONTROLLER.

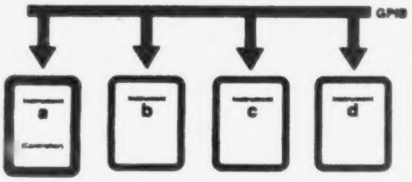
MAXIMUM CABLE LENGTH OF 20 METERS OR TWO METERS OF CABLE PER DEVICE ON THE BUS.
EXAMPLE:
3 DEVICES ON BUS - MAXIMUM CABLE LENGTH IS 2 METERS X 3 DEVICES = 6 METERS. THUS TWO DEVICES MAY HAVE 1 METER CABLES AND THE THIRD A 4 METER CABLE.

DEVICES ACTIVE ON THE BUS:
NORMALLY UP TO 1/3 OF THE DEVICES CONNECTED TO THE BUS MAY BE OFF. THIS MEANS THAT SOME DEVICES, EVENTHOUGH NOT CURRENTLY BEING USED, MUST BE ON.

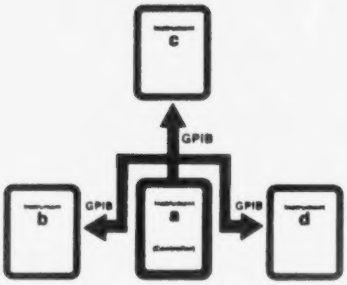
STACKING OF CONNECTORS:
LIMIT TO 3, FOR STRENGTH REASONS.

CONNECTION ARRANGEMENT:
LINEAR, STAR, OR COMBINATION

DIAGRAM COURTESY OF TEKTRONIX, INC. REPRINTED WITH PERMISSION



Linear Configuration



Star Configuration

NOTES:

INTRODUCTION: OVERVIEW OF THE IEEE 488 STANDARD

BASIC HARDWARE CONSIDERATIONS CONTINUED

DEVICE FUNCTIONS

LISTENER - A DEVICE THAT ACCEPTS INFORMATION OVER THE INTERFACE WHEN ADDRESSED (SELECTED).
EXAMPLES: PRINTERS, SIGNAL SOURCES, DMM RECEIVING SET UP INFORMATION. UP TO 14 DEVICES MAY BE LISTENING AT ONCE.

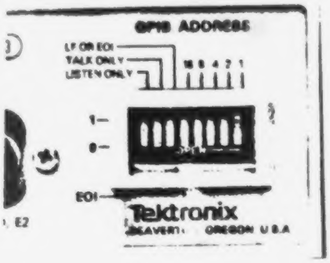
TALKER - A DEVICE THAT SENDS INFORMATION OVER THE INTERFACE WHEN ADDRESSED (SELECTED).
EXAMPLES: DMM SENDING A READING, DATA STORAGE DEVICES (DISKS) SENDING DATA. ONLY ONE DEVICE MAY BE A TALKER ON THE BUS AT ANY POINT IN TIME.

CONTROLLER - A DEVICE (USUALLY A COMPUTER) WHICH DECIDES WHICH OTHER DEVICES CONNECTED TO THE BUS WILL TALK AND LISTEN, AND WHEN. ONLY ONE CONTROLLER MAY BE ACTIVE ON THE BUS AT ANY GIVEN POINT IN TIME.

ADDRESSING

EACH DEVICE HAS A UNIQUE ADDRESS (0 TO 31) SET BY DIP SWITCHES ON (IN) THE INSTRUMENT.
NOTE: ADDRESS 31 IS GENERALLY NOT USED, THIS INSTRUCTS ALL DEVICES ON THE BUS TO STOP TALKING AND LISTENING.

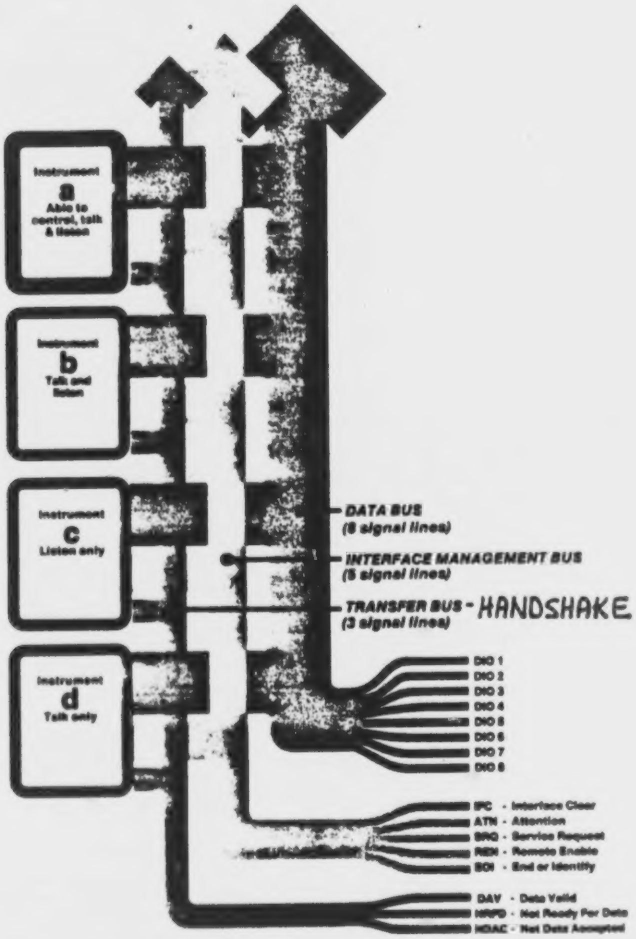
DIAGRAM COURTESY OF TEKTRONIX, INC. REPRINTED WITH PERMISSION



NOTES:

INTRODUCTION: OVERVIEW OF THE IEEE 488 STANDARD

BASIC HARDWARE CONSIDERATIONS CONTINUED


Instrument
a
Able to control, talk & listen
Instrument
b
Talk and listen
Instrument
c
Listen only
Instrument
d
Talk only
DATA BUS
(8 signal lines)
INTERFACE MANAGEMENT BUS
(5 signal lines)
TRANSFER BUS - HANDSHAKE
(3 signal lines)
DIO 1
DIO 2
DIO 3
DIO 4
DIO 5
DIO 6
DIO 7
DIO 8
IFC - Interface Clear
ATN - Attention
SRQ - Service Request
REN - Remote Enable
EOI - End or Identify
DAV - Data Valid
NRFD - Not Ready For Data
NDAC - Not Data Accepted

NOTES:

INTRODUCTION: GETTING STARTED

SYSTEM FAMILIARIZATION

NEW SYSTEM - WHEN A NEW SYSTEM HAS BEEN DECIDED UPON, REQUEST THE OPERATORS MANUALS IN ADVANCE TO BECOME FAMILIAR WITH THE SYSTEM.

START WITH THE CONTROLLER (COMPUTER)

- IF A NEW SYSTEM, START WITH INITIAL SETUP SECTION OF THE OPERATORS MANUAL.
- IN ADDITION TO GENERAL INFORMATION ON OPERATING PROCEDURES, MANY OPERATORS MANUALS CONTAIN TUTORIAL OR ABRIDGED OPERATING INSTRUCTIONS DESIGNED FOR RAPID SYSTEM LEARNING. UTILIZE THEM.
- KEY ELEMENTS TO LEARN
  - IS THE OPERATING SYSTEM ROM OR DISK BASED?
  - IF DISK BASED, WHAT IS LOADING PROCEDURE?
  - IS APPLICATIONS LANGUAGE ROM OR DISK BASED?
  - IF DISK BASED, WHAT IS LOADING PROCEDURE?
  - KEYBOARD OPERATION AND SPECIAL FUNCTION KEYS - TYPING AIDS?
  - IF MULTI DISK SYSTEM - HOW TO SELECT DESIRED DISK.

NOTES:

INTRODUCTION: GETTING STARTED

SYSTEM FAMILIARIZATION CONTINUED

ADD THE PERIPHERALS

IN MANY CASES HARD COPY OUTPUT (PRINTED) FROM A TEST IS REQUIRED i.e. A PRINTED COPY OF GATHERED DATA OR A GRAPH PLOT OF GAIN V.S. FREQUENCY.

PRINTERS - 3 INTERFACES

- IEE 488
- RS-232 (SERIAL)
- CENTRONICS (PARALLEL)

EACH INTERFACE CAN REQUIRE A DIFFERENT ACCESS PROTOCOL i.e. SEQUENCE OF COMMANDS NECESSARY TO SEND IT INFORMATION.

DOES THE COMPUTER SUPPORT (OR HAVE INSTALLED) THE NECESSARY INTERFACE.

GENERIC PRINTERS (EPSON, OKIDATA, ETC.) MAY NOT SUPPORT SPECIAL CHARACTERS AND GRAPHICS.

FOR RS-232, THE COMPUTER AND PRINTER MUST BE SET TO THE SAME OPERATING PROTOCOLS i.e. PARITY, BAUD RATE, # START BITS, # DATA BITS, ETC.

FOR IEEE 488, THE PRINTER MUST HAVE A UNIQUE ADDRESS. UTILIZE THE FACTORY DEFAULT IF THIS ADDRESS IS NOT UTILIZED BY OTHER DEVICES. A SYSTEM MAY HAVE MORE THAN ONE PRINTER - EACH WITH ITS OWN ADDRESS.

NOTES:

INTRODUCTION: GETTING STARTED

SYSTEM FAMILIARIZATION CONTINUED

ADD THE PERIPHERALS CONTINUED

PLOTTERS - SAME REQUIREMENTS AS PRINTERS

USUALLY RS-232 OR IEEE 488

FOR IEEE 488, MAY HAVE SECONDARY ADDRESS i.e. ONE TO LIFT AND ONE TO LOWER THE PEN (MORE ON SECONDARY ADDRESSES LATER).

MASS STORAGE DEVICES

THESE MAY HAVE A SPECIAL INTERFACE i.e. IBM PC, FLUKE 1720 OR UTILIZE THE GPIB.

IF SYSTEM UTILIZES MULTIPLE MASS STORAGE DEVICES, EACH IS ACCESSED BY ITS OWN COMMAND OR ADDRESS.

A HARD DISK MAY BE SET UP TO LOOK LIKE SEVERAL FLOPPY DISKS, EACH WITH ITS OWN ACCESS CODE OR ADDRESS.

EDISK OR RAMDISK - A LARGE AREA OF RAM WHICH IS ACCESSED AS IF IT WERE A DISK, VERY FAST.

NOTES:

SYSTEM FAMILIARIZATION CONTINUED

ADD THE PERIPHERALS CONTINUED

TEST & MEASUREMENT EQUIPMENT

ITEMS WHICH FALL INTO THIS CLASSSIFICATION INCLUDE: VOLTMETERS (DMM), SIGNAL SOURCES, MATRIX SWITCHES, PROGRAMMABLE POWER SOURCES, COUNTERS, OSCILLOSCOPES, ETC.

DEPTH OF COMPATABILITY - JUST BECAUSE AN INSTRUMENT IS ADVERTISED AS IEEE 488 COMPATABLE, DOES NOT MEAN THAT ALL OF ITS FUNCTIONS ARE PROGRAMMABLE VIA THE CONTROLLER. EXAMPLE: A KEITHLY 179A's FUNCTION AND RANGE ARE NOT BUS PROGRAMMABLE, WHERE AS THE FLUKE 8840 AND HP 3478 ARE.

SOME INSTRUMENTS HAVE MULTIPLE PRIMARY (AS COMPARED TO SECONDARY) ADDRESSES. ON THESE TYPES OF INSTRUMENTS YOU SET ONE ADDRESS AND THE NEXT HIGHER ADDRESS IS THE SECOND ADDRESS, AND SO ON.

INSTRUMENTS WITH DISPLAYS (DMMs, COUNTERs, ETC.) CAN BE QUERIED AS TO THEIR ADDRESS SETTING BY A FRONT PANEL BUTTON OR AT POWER UP.

FOR BEST RESULTS ONLY CHANGE A DEVICES ADDRESS WHEN IT IS POWERED DOWN.

MESSAGES REQUIRE A TERMINATOR, TO SIGNIFY THE END OF A MESSAGE. MAKE SURE THE CONTROLLER AND INSTRUMENTS USE THE SAME TERMINATOR (IT MAY BE CHANGED ON MANY DEVICES. COMMON TERMINATORS: CARRIAGE RETURN - Cr & LINE FEED - Lf.).

NOTES:

INTRODUCTION: GETTING STARTED

SYSTEM FAMILIARIZATION CONTINUED

ADD THE PERIPHERALS CONTINUED

GPIB INTERFACE

INCLUDED ON HP 200 SERIES, 86, 87; FLUKE 1720 SERIES

OPTIONAL FOR IBM PC - PLUG IN INTERFACE CARDS BY HP, NATIONAL INSTRUMENTS. FOR NON-IBM TYPE COMPUTERS I/O TECH HAS AN RS-232 TO GPIB INTERFACE.

PLUG IN ADAPTER FOR HP 85.

INTERFACE OPTIONS ARE ALSO AVAILABLE FOR A WIDE RANGE OF MINI COMPUTERS.

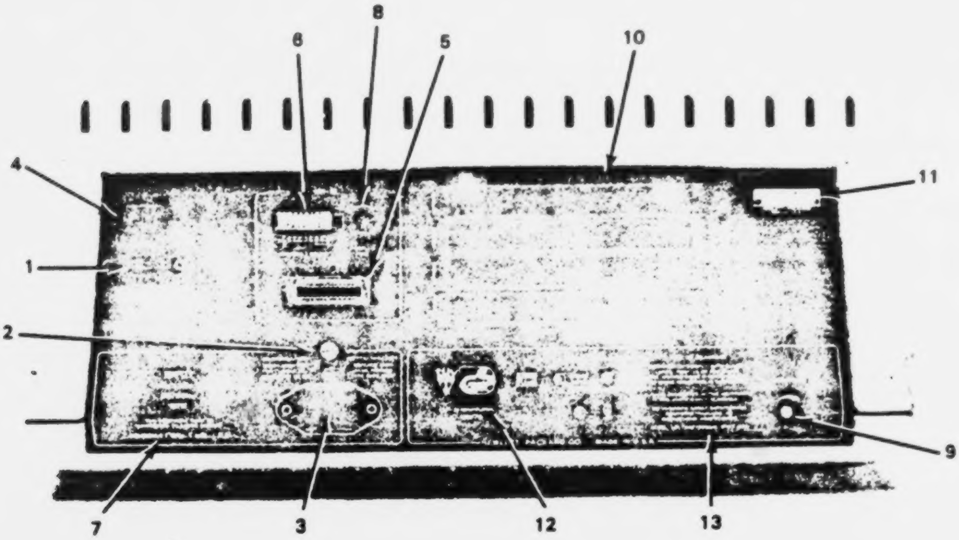
MANY COMPUTERS CAN HANDLE MULTIPLE GPIB INTERFACES. EACH HAS ITS' OWN SELECT CODE. AN INSTRUMENT ON ONE INTERFACE MAY HAVE THE SAME ADDRESS AS AN INSTRUMENT ON ANOTHER INTERFACE, BUT THE SELECT CODES WILL BE DIFFERENT.

NOTES:

INTRODUCTION: LAB FAMILIARIZATION

THE FOLLOWING MATERIAL IS BASED ON THE FOLLOWING HEWLWTT PACKARD PRODUCTS: HP 87 COMPUTER, HP 3478A DMM, HP 3314A FUNCTION GENERATOR. ADEQUATE INFORMATION WILL BE INCLUDED TO ALLOW THOSE WHO DO NOT HAVE ACCESS TO THIS EQUIPMENT, TO GAIN AN USEABLE BACKGROUND.

THE CONTROLLER, DIAGRAM COURTESY OF HEWLETT PACKARD CO., REPRINTED WITH PERMISSION



THE HP 87 HAS ONE BUILT IN GPIB INTERFACE (#5). MORE MAY BE ADDED IN THE EXPANSION AREA (#10). THE SELECT CODE AND ADDRESS FOR THE BUILT IN INTERFACE (#6) MUST BE UNIQUE. USE FACTORY PRESET VALUE, WHICH IS 7.

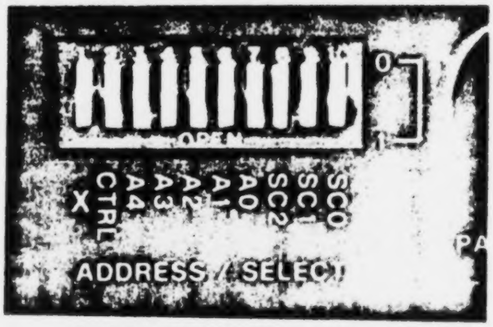
DIAGRAM COURTESY OF HEWLETT PACKARD CO., REPRINTED WITH PERMISSION

SELECT CODE

SC2	SC1	SC0
1	0	0

ADDRESS CODE

A4	A3	A2	A1	A0
1	0	1	0	1



SWITCH SEGMENT 2 SHOULD BE SET TO 1 FOR THE 87 TO ACT AS THE CONTROLLER

DETAILS ON Pgs 46-48 HP87 INTRODUCTORY MANUAL

POWER SWITCH (#4) IS ON THE REAR PANEL.

NOTES:

INTRODUCTION: LAB FAMILIARIZATION CONTINUED

POWER UP

TURN ON THE DISK DRIVE, THEN THE COMPUTER- POWER SWITCH ON REAR

ON POWER UP THE COMPUTER PERFORMS A SELF TEST, WHEN COMPLETE THE CURSOR APPEARS IN THE UPPER LEFT CORNER.

THE AMBER LIGHT AT THE BOTTOM OF THE KEYBOARD INDICATES POWER ON CONDITION.

CHECK WITH YOUR INSTRUCTOR IF YOU GET AN ERROR OR WARNING MESSAGE.

YOU ARE NOW READY TO START.

LOADING LAB1-1

INSERT THE DISK AS SHOWN, COURTESY OF HEWLETT PACKARD CO., REPRINTED WITH PERMISSION

82901M FLEXIBLE DISC DRIVE

TYPE IN THE FOLLOWING COMMAND

```
LOAD "LAB1-1"
```

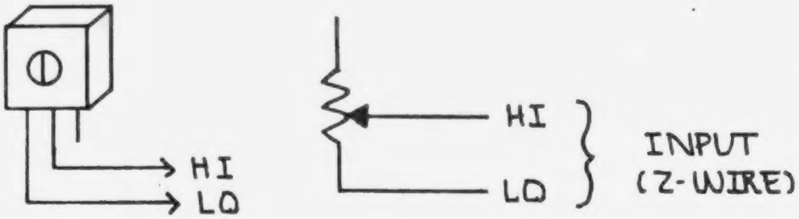
PRESS THE **END LINE** KEY

YOU ARE NOW READY TO BEGIN YOUR FIRST LAB EXERCISE

NOTES:

RUNNING LAB1-1

CONNECT A 10K OHM POTENTIOMETER TO THE 3478A DMM AS SHOWN BELOW. SET THE POTENTIOMETER TO 5K OHMS.



WATCH THE DMM DISPLAY IMMEDIATELY AFTER YOU PRESS THE **RUN** KEY.

TAKE TIME TO RUN LAB1-1 AND ANSWER THE FOLLOWING QUESTIONS. ANSWERS IN APPENDIX A.

1. WHAT MESSAGE IS DISPLAYED ON THE DMM DISPLAY?

TURN THE POTENTIOMETER FULLY CLOCKWISE.

2. WHAT MESSAGE IS NOW DISPLAYED ON THE DMM DISPLAY?

TURN THE POTENTIOMETER FULLY COUNTERCLOCKWISE.

3. WHAT MESSAGE IS NOW DISPLAYED ON THE DMM DISPLAY?

TURN TO THE PROGRAM LISTING ON THE NEXT PAGE, STUDY THE OPERATION OF THE PROGRAM, AND ANSWER THE QUESTIONS ON PAGE 1-17

NOTES:

```
10 !    LAB1-1: POTENTIOMETER CALIBRATION PROGRAM
20 !    ----------------------------------------
30 !
40 !    CONNECT A 10K POTENTIOMETER TO THE HP 3478A DMM,
50 !    THEN FOLLOW DIRECTIONS ON THE DMM DISPLAY.
60 !
70 !                                                             COMMENTS
80 CLEAR !                                                    --------
90 OUTPUT 723 ;"F3R4" !                            SETS UP DMM TO OHMS, 30K RANGE
100 OUTPUT 723 ;"D2THIS IS A CO" !                 SENDS MESSAGE FOLLOWING
110 !                                              D2 TO DMM DISPLAY
120 WAIT 250 !                                     MESSAGE IS DISPLAYED 0.25 SEC
130 OUTPUT 723 ;"D2HIS IS A COM"
140 WAIT 250
150 OUTPUT 723 ;"D2IS IS A COMP"
160 WAIT 250
170 OUTPUT 723 ;"D2S IS A COMPU"
180 WAIT 250
190 OUTPUT 723 ;"D2 IS A COMPUT"
200 WAIT 250
210 OUTPUT 723 ;"D2IS A COMPUTE"
220 WAIT 250
230 OUTPUT 723 ;"D2S A COMPUTER"
240 WAIT 250
250 OUTPUT 723 ;"D2 A COMPUTER "
260 WAIT 250
270 OUTPUT 723 ;"D2A COMPUTER R"
280 WAIT 250
290 OUTPUT 723 ;"D2 COMPUTER RU"
300 WAIT 250
310 OUTPUT 723 ;"D2COMPUTER RUN"
320 WAIT 250
330 OUTPUT 723 ;"D2OMPUTER RUN "
340 WAIT 250
350 OUTPUT 723 ;"D2MPUTER RUN T"
360 WAIT 250
370 OUTPUT 723 ;"D2PUTER RUN TE"
380 WAIT 250
390 OUTPUT 723 ;"D2UTER RUN TES"
400 WAIT 250
410 OUTPUT 723 ;"D2TER RUN TEST"
420 WAIT 250
430 OUTPUT 723 ;"D2ER RUN TEST "
440 WAIT 250
450 OUTPUT 723 ;"D2R RUN TEST  "
460 WAIT 250
470 OUTPUT 723 ;"D2 RUN TEST   "
480 WAIT 250
490 OUTPUT 723 ;"D2INSERT PART"
500 WAIT 250
510 OUTPUT 723 ;"F3R4" !                           SET DMM TO OHMS, RANGE 30K
520 ENTER 723 ; S1 !                               DMM SENDS OHM READING TO
530 !                                              COMPUTER
540 IF S1>5500 OR S1<4800 THEN 550 ELSE 100 !      CHECKS VALUE, IN SPEC ?
550 IF S1>5500 THEN 560 ELSE 590 !                 VALUE HI ?, LO ?
560 OUTPUT 723 ;"D2TURN CCLKWSE" !                 CORRECTION MESSAGE
570 WAIT 1000 !                                    HOLD MESSAGE FOR 1 SEC.
580 GOTO 510 !                                     REPEAT PROCESS
590 OUTPUT 723 ;"D2TURN CLKWISE" !                 CORRECTION MESSAGE
```

```
600 WAIT 1000 !                                     HOLD MESSAGE FOR 1 SEC.
610 GOTO 510
620 END
```

1-16

INTRODUCTION: LAB EXERCISES CONTINUED

LAB1-1 ANALYSIS

4. WHAT IS THE ADDRESS OF THE DMM?

5. IF YOU DO NOT KNOW THE DMM's ADDRESS, WHAT ARE TWO WAYS YOU CAN DETERMINE IT.

6. WHAT DO YOU DO IF YOU NEED TWO DMMs ON THE BUS?

7. WHICH LINE STATEMENT SETS UP THE DMM TO READ OHMS?

8. IS THE DMM SETUP TO AUTO RANGE OR IS IT SET ON A PARTICULAR RANGE?

9. EXPLAIN HOW THE MESSAGE IS SCROLLED ACROSS THE DMM's DISPLAY.

10. IF THE "CALIBRATED" POSITION OF THE POTENTIOMETER IS 5K OHMS, WHAT POSITIVE TOLERANCE IS ALLOWED_____ AND WHAT NEGATIVE TOLERANCE IS ALLOWED_____?

NOTES:

INTRODUCTION: LAB EXERCISES CONTINUED

LAB1-1 MODIFICATION (LAB1-1a)

YOUR SUPERVISOR HAS DECIDED THAT THE PROGRAM MUST BE MODIFIED. ONCE EACH POTENTIOMETER HAS BEEN CALIBRATED ITS OHMIC VALUE IS TO BE AUTOMATICALLY OUTPUT TO A PRINTER.

ONE WAY TO ACCOMPLISH THIS IS TO OUTPUT THE DMM READING TO THE PRINTER WHENEVER THE POTENTIOMETER VALUE IS IN THE CALIBRATED RANGE. TRY THIS BY MAKING THE FOLLOWING MODIFICATIONS: ADD - 91 GOTO 100, 92 OUTPUT 705;" CALIBRATED VALUE IS ",S1 , THE 100 IN LINE 540 MUST BE CHANGED TO 92. **MAKE THESE MODIFICATIONS & RUN IT**

UNFORTUNATELY IF THE PERSON PERFORMING THE CALIBRATION ADJUSTS THE POTENTIOMETER IN AND OUT OF THE TOLERANCE RANGE, DIFFERENT VALUES WILL BE OUTPUT TO THE PRINTER. AS LONG AS THE POTENTIOMETER IS WITHIN TOLERANCE, ITS' VALUE WILL BE SENT TO THE PRINTER.

THE COMPUTER MUST WAIT UNTIL THE OPERATOR HAS COMPLETED THE CALIBRATION, AND THEN PRINT ONE READING.

THE DMM HAS A FRONT PANEL BUTTON LABELED SRQ. THIS STANDS FOR SERVICE REQUEST. IF THE COMPUTER IS PROGRAMMED TO RESPOND TO SRQs, THEN IT MAY BE USED TO TELL THE COMPUTER WHEN TO PRINT THE VALUE i.e. PRINT VALUE WHEN OPERATOR PRESSES SRQ.

READ PAGES 12-14 IN HP PRODUCT NOTE 3478A-1 "USING THE 3478A". ENTER THE FOLLOWING STATEMENTS IN THEIR APPROPRIATE LOCATIONS IN PROGRAM LAB1-1. BE SURE TO CHANGE LINE 100 BACK TO ITS ORIGIONAL VALUE PRIOR TO THE NEXT SET OF MODIFICATIONS.

```
91 OUTPUT 723;"KM20"
92 ON INTR 7 GOSUB 1000
93 ENABLE INTR 7;8
1000 STATUS 7,1;A
1010 P=SPOLL (723)
1020 IF BIT (P,4  THEN OUTPUT 705;" CALIBRATED VALUE IS ",S1
1030 RETURN
```

DETAILS OF SERVICE REREQUEST PROGRAMMING WILL BE COVERED LATER.

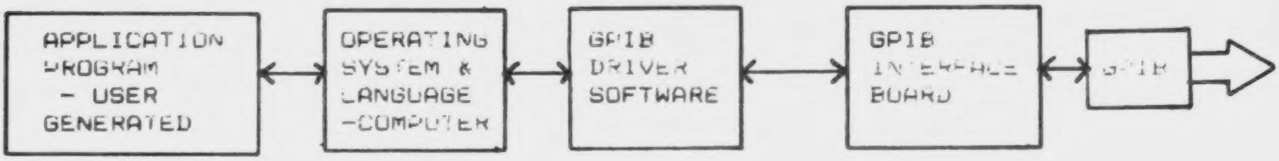
NOTES:

```
10 !    LAB1-1a POTENTIOMETER CALIBRATION PROGRAM
20 !    ----------------------------------------
30 !
40 !    CONNECT A 10K POTENTIOMETER TO THE HP 3478A DMM,
50 !    THEN FOLLOW DIRECTIONS ON THE DMM DISPLAY.
60 !
70 !                                                         COMMENTS
80 CLEAR !                                                    --------
90 OUTPUT 723 ;"F3R4" !                              SETS UP DMM TO OHMS, 30K RANGE
91 OUTPUT 723 ;"KM20"
92 ON INTR 7 GOSUB 1000
93 ENABLE INTR 7;8
100 OUTPUT 723 ;"D2THIS IS A CO" !                   SENDS MESSAGE FOLLOWING
110 !                                                D2 TO DMM DISPLAY
120 WAIT 250 !                                       MESSAGE IS DISPLAYED 0.25 SEC
130 OUTPUT 723 ;"D2HIS IS A COM"
140 WAIT 250
150 OUTPUT 723 ;"D2IS IS A COMP"
160 WAIT 250
170 OUTPUT 723 ;"D2S IS A COMPU"
180 WAIT 250
190 OUTPUT 723 ;"D2 IS A COMPUT"
200 WAIT 250
210 OUTPUT 723 ;"D2IS A COMPUTE"
220 WAIT 250
230 OUTPUT 723 ;"D2S A COMPUTER"
240 WAIT 250
250 OUTPUT 723 ;"D2 A COMPUTER "
260 WAIT 250
270 OUTPUT 723 ;"D2A COMPUTER R"
280 WAIT 250
290 OUTPUT 723 ;"D2 COMPUTER RU"
300 WAIT 250
310 OUTPUT 723 ;"D2COMPUTER RUN"
320 WAIT 250
330 OUTPUT 723 ;"D2OMPUTER RUN "
340 WAIT 250
350 OUTPUT 723 ;"D2MPUTER RUN T"
360 WAIT 250
370 OUTPUT 723 ;"D2PUTER RUN TE"
380 WAIT 250
390 OUTPUT 723 ;"D2UTER RUN TES"
400 WAIT 250
410 OUTPUT 723 ;"D2TER RUN TEST"
420 WAIT 250
430 OUTPUT 723 ;"D2ER RUN TEST "
440 WAIT 250
450 OUTPUT 723 ;"D2R RUN TEST  "
460 WAIT 250
470 OUTPUT 723 ;"D2 RUN TEST   "
480 WAIT 250
490 OUTPUT 723 ;"D2INSERT PART"
500 WAIT 250
510 OUTPUT 723 ;"F3R4" !                             SET DMM TO OHMS, RANGE 30K
520 ENTER 723 ; S1 !                                 DMM SENDS OHM READING TO
530 !                                                COMPUTER
540 IF S1>5500 OR S1<4800 THEN 550 ELSE 92 !        CHECKS VALUE, IN SPEC ?
550 IF S1>5500 THEN 560 ELSE 590 !                   VALUE HI ?, LO ?
560 OUTPUT 723 ;"D2TURN CCLKWSE" !                   CORRECTION MESSAGE
```

```
570 WAIT 1000 !                             HOLD MESSAGE FOR 1 SEC.
580 GOTO 510 !                              REPEAT PROCESS
590 OUTPUT 723 ;"D2TURN CLKWISE" !          CORRECTION MESSAGE
600 WAIT 1000 !                             HOLD MESSAGE FOR 1 SEC.
610 GOTO 510 !                              REPEAT
620 END
1000 STATUS 7,1 ; A
1010 P=SPOLL (723)
1020 IF BIT (P,4) THEN OUTPUT 705 ;" CALIBRATED VALUE IS ",S1
1030 RETURN
1040 !  NOTES:
```

HARDWARE: THE CONTROLLER

PRELUDE - THE CONTROLLER (COMPUTER) IS THE HEART OF THE TEST SYSTEM. ITSFUNCTION IS TO INITIATE OPERATIONS BY THE INSTRUMENTS IN THE SYSTEM, AND TO INTERPERT THE INFORMATION GATHERED. TWO GENERAL TYPES OF CONTROLLERS MAY BE USED IN TEST SYSTEMS. THE FIRST TYPE IS ONE THAT HAS BEEN DESIGNED AS AN INSTRUMENT CONTROLLER. THIS COMPUTER HAS AN OPERATING SYSTEM THAT IS DESIGNED TO IMPLEMENT ALL GPIB OPERATIONS, SUCH AS: ADDRESSING INSTRUMENTS, SEND DEVICE DEPENDENT AND INTERFACE MESSAGES, TRANSMITT AND RECEIVE DATA, AND BE ABLE TO RESPOND TO INTERRUPTS (SRQs). THE SOFTWARE IS HIGH LEVEL WITH THE SPECIAL COMMANDS REQUIRED FOR GPIB OPERATIONS. EXAMPLES OF THIS TYPE OF CONTROLLER ARE: HP SERIES 80 & 200 COMPUTERS, FLUKE 1700 SERIES, AND TEKTRONIX 4041. THE SECOND TYPE OF CONTROLLER IS ONE THAT UTILIZES A GENERIC COMPUTER WITH AN ADD ON BOARD OR BOX TO MAKE IT ACT AS A GPIB CONTROLLER. COMPUTERS SUCH AS THE IBM PC AND APPLE FALL INTO THIS CATAGORY. SINCE THESE WERE NOT DESIGNED AS INSTRUMENT CONTROLLERS, THEY HAVE LIMITATIONS. SOME OF THESE LIMITATIONS ARE SOFTWARE WHICH IS LESS USER FRIENDLY, SPEED, NOT IMPLEMENTING ALL GPIB FUNCTIONS, NO REAL TIME INTERRUPT (SRQ) PROCESSING. THE KEY ADVANTAGE IS COST.


APPLICATION PROGRAM - USER GENERATED
OPERATING SYSTEM & LANGUAGE -COMPUTER
GPIB DRIVER SOFTWARE
GPIB INTERFACE BOARD
GPIB

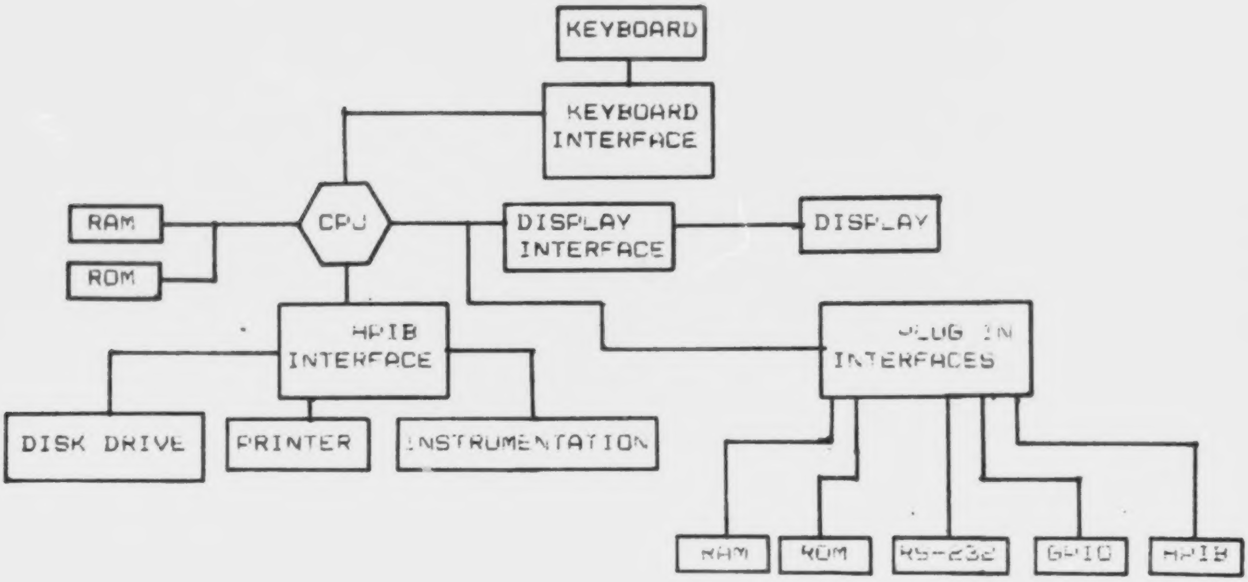
3 LEVELS OF SOFTWARE REQUIRED

NOTES:

HARDWARE: THE CONTROLLER

THE HP 86/87 HAS BEEN CHOSEN AS A REPRESENTATIVE CONTROLLER. THE CONCEPTS LEARNED HERE MAY BE APPLIED TO OTHER CONTROLLERS SINCE THEIR OPERATING PROCEDURES ARE SIMILAR.

FUNCTIONAL BLOCK DIAGRAM


KEYBOARD
KEYBOARD INTERFACE
RAM
ROM
CPU
DISPLAY INTERFACE
DISPLAY
HPIB INTERFACE
PLUG IN INTERFACES
DISK DRIVE
PRINTER
INSTRUMENTATION
RAM
ROM
RS-232
GPIO
HPIB

CPU - CUSTOM 8 BIT

GPIB (HPIB) - BUILT IN, MAY ADD ADDITIONAL

RAM - MAY UPGRADE TO 640K/512K

ROM - INTERNAL OPERATING SYSTEM, EXTERNAL I/O ENHANCEMENTS & ADVANCED PROGRAMMING CAPABILITIES.

PLUG IN INTERFACES - (PARTIAL LISTING)
RS232 SERIAL PORT
GPIO - GENERAL PURPOSE I/O - 16 BIT
PRINTER PARALLEL PORT - CENTRONICS TYPE

NOTES:

HARDWARE: THE CONTROLLER

KEYBOARD: KEY FUNCTIONS

ALPHANUMERIC KEYS -

LAYOUT - QWERTY, TYPEWRITER TYPE KEYBOARD

UPSHIFTED LETTERS ARE UPPERCASE!!!

LOWERCASE

SHIFT KEY - MOMENTARY ACCESS

CAPS LOCK - EXTENDED ACCESS, ONLY FOR ALPHAKEYS

REPEATING KEYS: ALL ALPHANUMERIC (THIS INCLUDES SYMBOLS & SPACE BAR)

CTRL KEY (CONTROL) - FOR ALTERNATE CHARACTERS & CONTROL CODES. EXAMPLE: CTRL O DISPLAYS THE PI SYMBOL. THE FOLLOWING KEY ENTRY: CHR$(27)&"&a15" MAY BE ENTERED AS CTRL{&"&a15L" i.e. CHR$(27) = CTRL {. WHEN SENT TO A PRINTER CHR$(27) IS A CODE CALLED ESC (ESCAPE) WHICH TELLS THE PRINTER THE INFORMATION FOLLOWING ARE COMMANDS AND SHOULD NOT BE PRINTED. THESE COMMANDS WILL SET THE LEFT MARGIN TO COLUMN 50.

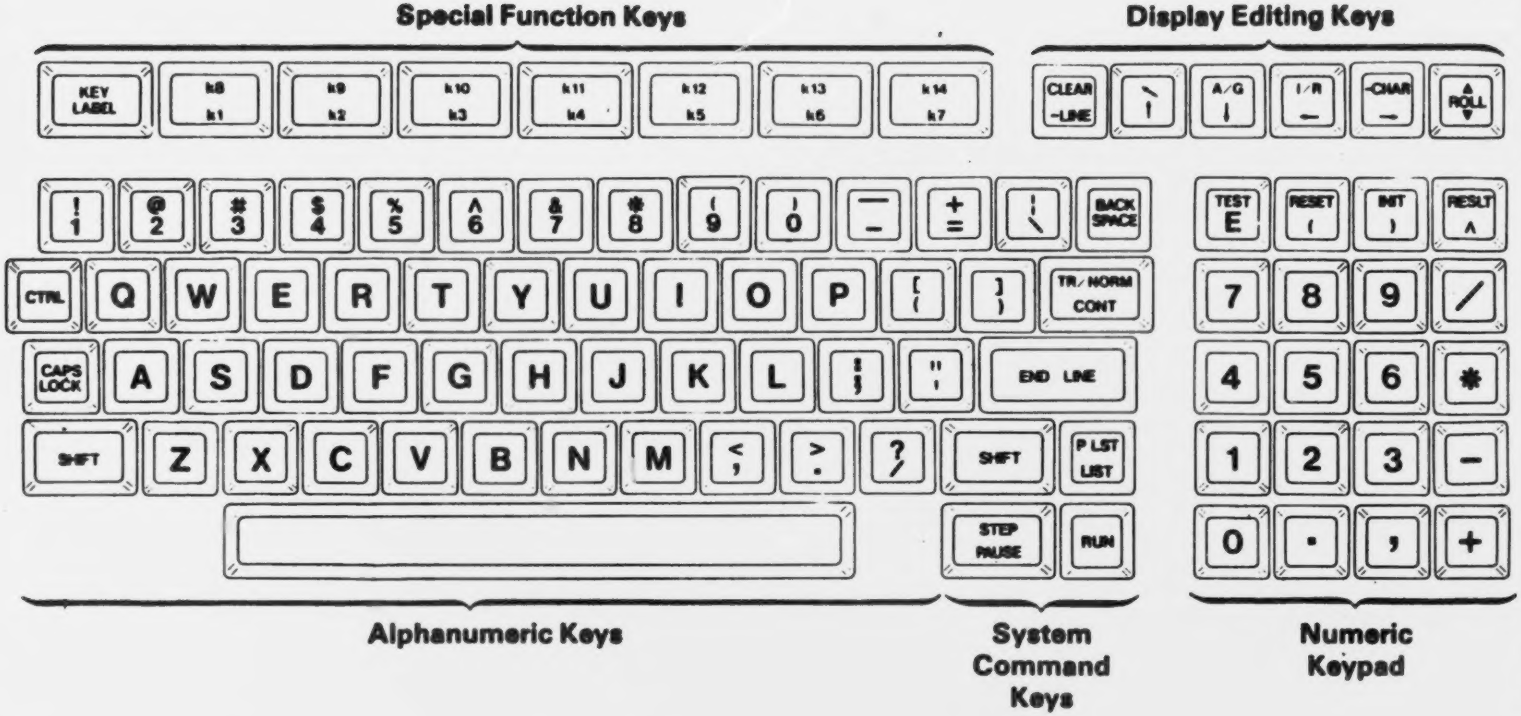
TO CONVERT THE KEYBOARD SO UNSHIFTED CHARACTERS ARE LOWERCASE, TYPE THE FLIP STATEMENT, THEN PRESS ENDLINE.

NOTES:

HARDWARE: THE CONTROLLER

KEYBOARD LAYOUT

COURTESY OF HEWLETT PACKARD, REPRINTED WITH PERMISSION


Special Function Keys
Display Editing Keys
KEY LABEL
k8 k1
k9 k2
k10 k3
k11 k4
k12 k5
k13 k6
k14 k7
CLEAR -LINE
A/G
I/R
-CHAR
ROLL
BACK SPACE
TEST E
RESET (
INIT)
RESLT ^
CTRL
Q W E R T Y U I O P
TR/NORM CONT
CAPS LOCK
A S D F G H J K L
END LINE
SHIFT
Z X C V B N M
P LST LIST
STEP PAUSE
RUN
7 8 9 /
4 5 6 *
1 2 3 −
0 . , +
Alphanumeric Keys
System Command Keys
Numeric Keypad

KEYBOARD: KEY FUNCTIONS

SYSTEM COMMAND KEYS

ENDLINE - ENTERS A STATEMENT INTO MEMORY, CALLED RETURN ON OTHER COMPUTERS.

LIST - DISPLAYS LISTING OF CURRENT PROGRAM IN MEMORY ON CRT.

P LIST - PRINTS LISTING ON PRINTER, IF ON & COMPUTER KNOWS PRINTERS ADDRESS, OTHERWISE IT IS LISTED ON CRT.

RUN - EXECUTES PROGRAM IN MEMORY.

PAUSE - STOPS RUNNING PROGRAM, PROGRAM MAY BE LISTED AND MODIFIED.

STEP - EXECUTES PROGRAM LINE BY LINE WHEN IN PAUSE. IF PROGRAM HAS BEEN MODIFIED IN PAUSE IT MUST BE INITIALIZED PRIOR TO STEP. SEE MANUAL.

CONT - RESUMES EXECUTION OF PAUSED PROGRAM. IF EDITED DURING PAUSE IT MUST BE INITIALIZED. SEE MANUAL.

INIT - INITIALIZES PROGRAM IN MEMORY. THIS SETS PROGRAM VARIABLES TO UNDEFINED VALUES & WILL SET PROGRAM POINTER TO BEGINNING.

RESET - RETURNS COMPUTER TO POWER ON STATE, DOES NOT ERASE MEMORY.

TR/NORM - FOR INITIATING AND CANCELLING TRACE ALL OPERATION. LISTS PROGRAM BRANCH & RETURN LINE NUMBERS & VARIABLE VALUES.

NOTES:

HARDWARE: THE CONTROLLER

KEYBOARD: KEY FUNCTIONS

NUMERIC KEYPAD

DIGITS 0-9 & DECIMAL POINT

ARITHMETIC OPERATORS +, -, *, /

EXPONENTIATION - E

PARENTHESIS ()

SPECIAL FUNCTION KEYS

K1-K7 UNSHIFTED

K8-K14 SHIFTED

IN CALCULATOR MODE (NO PROGRAM RUNNING - POWER LIGHT NOT FLASHING). THEY ACT AS TYPING AIDS.

K1	AUTO	K8	TRACE VAR
K2	DISP	K9	CRT IS
K3	PRINT	K10	PRINTER IS
K4	MASS STORAGE IS	K11	INITIALIZE
K5	DELETE	K12	SCRATCH
K6	LOAD	K13	LOADBIN
K7	STORE	K14	STOREBIN

IN PROGRAM MODE (PROGRAM RUNNING - POWER LIGHT BLINKING). ACTIVATING THEM WILL CAUSE THE PROGRAM TO BRANCHTO A LOCATION PREVIOUSLY DEFINED USING THE STATEMENT ON KEY #.

EXAMPLE:

```
10 ON KEY # 1 GOSUB 100
20 ON KEY # 2 GOSUB 200
30 GOTO 10
.
100 SUBROUTINE STARTS HERE
.
1XX SUBROUTINE ENDS HERE
1XY RETURN
200 SUBROUTINE STARTS HERE
.
XXX SUBROUTINE ENDS HERE
XXY RETURN
```

KEY LABEL - DISPLAYS LABELS OF SPECIAL FUNCTION KEYS ASSIGNED BY USER.

SHIFT KEY LABEL - DISPLAYS LABELS OF TYPING AIDS.

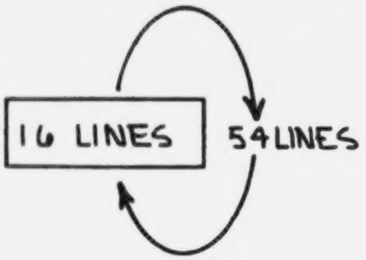
NOTES:

HARDWARE: THE CONTROLLER

DISPLAY FORMAT

16 LINES X 80 CHARACTERS

SCROLLING ACCESS TO 54 LINES USING THE **ROLL** KEY



24 LINES MAY BE DISPLAYED BY ENTERING PAGESIZE 24, TO RETURN TO 16 LINES ENTER PAGESIZE 16.

DISPLAY: EDITING

CLEAR - CLEARS INFORMATION ON DISPLAY ONLY

-LINE - CLEARS FROM CURSOR TO END OF LINE

I/R - INSERT REPLACE, DOUBLE CURSOR SHOWS WHERE CHARACTERS MAY BE INSERTED

-CHAR - DELETE A CHARACTER, REMAINING CHARACTERS ARE SHIFTED LEFT!

BACKSPACE - MOVES CURSOR TO LEFT AND ERASES CHARACTER AT NEW POSITION

SHIFT BACKSPACE - CLEARS LINE TO LEFT OF CURSOR

EXPRESSIONS UP TO 159 CHARACTERS MAY BE ENTERED PRIOR TO ENDLINE. THIS ALLOWS MULTIPLE EXPRESSIONS TO BE CONCATENATED PRIOR TO USING ENDLINE (CONCATENATION SYMBOL &)

DISPLAY: ADDRESSING

NORMALLY THE CRT DEVICE SELECTOR IS 1. THE STATEMENTS DISP, DISP USING, CAT, ERRORS, WARNINGS, AND LISTings ARE SENT TO THE DEVICE SELECTOR DEFINED IN THE "CRT IS" STATEMENT. FOR EXAMPLE, IF A PRINTER IS CONNECTED AT 705 AND A "CRT IS 705" IS EXECUTED, PERFORMING A CAT (CATALOG LISTING, ALSO KNOWN AS A DIRECTORY ON OTHER SYSTEMS) WILL OUTPUT THE DISK DIRECTORY TO THE PRINTER RATHER THAN THE CRT.

PRINTALL WILL PRINT ALL THE INFORMATION ON THE CRT SCREEN, CANCEL BY THE NORMAL STATEMENT.

NOTES:

HARDWARE: THE CONTROLLER

MASS STORAGE - THE HP 86/87 HAVE EXTERNALLY CONNECTED MASS STORAGE DEVICES. THESE COMMUNICATE WITH THE COMPUTER OVER THE GPIB BUS.

TYPES OF MASS STORAGE AVAILABLE

- FLEXIBLE (FLOPPY) DISK, 3.5" & 5.25" (SINGLE AND DUAL)
- HARD DISK 5 MBYTE & 10 MBYTE
- COMBINATION FLEXIBLE & FIXED DISK DRIVES

DEVICE ADDRESS (MASS STORAGE UNIT SPECIFIER - MSUS)

EACH DRIVE MUST HAVE A UNIQUE ADDRESS, DUAL DRIVES HAVE TWO

HARD DISK DRIVES MAY BE APPORTIONED INTO VOLUMES, EACH VOLUME WITH ITS OWN UNIQUE ADDRESS.

DEFAULT MASS STORAGE LOCATION

WHEN THE COMPUTER IS TURNED ON OR RESET, IT SEARCHES FOR THE DRIVE WITH THE LOWEST ADDRESS. THIS BECOMES THE **DEFAULT DRIVE.**

THE DEFAULT DRIVE IS THE ONE THAT IS CURRENTLY COMMUNICATING WITH THE COMPUTER.

IN ORDER FOR THIS TO OCCUR THE DRIVE MUST BE ON BEFORE THE COMPUTER IS TURNED ON, OR RESET.

TO CHANGE THE DEFAULT DRIVE (TO COMMUNICATE WITH ANOTHER DRIVE) THE MASS STORAGE STATEMENT MUST BE EXECUTED OR ALL DRIVE REFERENCED COMMANDS MUST SPECIFY THE NON-DEFAULT DRIVE.

EXAMPLE: ASSUME TWO DRIVES, ONE AT 700 THE OTHAR AT 710. EXECUTING THE STATEMENT MASS STORAGE IS ":D710" CHANGES THE DEFAULT DRIVE FROM 700 TO 710.

EXAMPLE: THE COMMAND LOAD "TEST PRG" WILL LOAD THE SPECIFIED PROGRAM FROM THE DEFAULT DRIVE. LOAD "TEST 2:D710" WILL LOAD FROM THE NON-DEFAULT DRIVE 710.

NOTES:

HARDWARE: THE CONTROLLER

MASS STORAGE CONTINUED

VOLUME LABELS

THIS IS A METHOD USED TO IDENTIFY A DISK. IN A MULTIPLE DRIVE SYSTEM A DISKS VOLUME LABEL MAY BE SPECIFIED AND THE COMPUTER WILL SEARCH UNTIL IT IS (IS NOT) FOUND. THIS ALLOWS ACCESSING A DISK DIRECTLY EVEN IF IT IS NOT IN THE DEFAULT DRIVE.

ASSIGNING VOLUME LABELS

```
VOLUME ":MSUS"                IS "NEW VOLUME LABEL"
       ".OLD VOLUME LABEL"
```

EXAMPLE: VOLUME ":D700" IS "FLDISK"

DISK INITIALIZATION

- CLEARS USED DISK
- TESTS DISK (FOR BAD TRACKS)
- SETS UP FILE DIRECTORY

ATTEMPTING TO CATALOG (CAT) AN UNINITIALIZED DISL WILL GENERATE A ERROR MESSAGE.

TO INITIALIZE A NEW DISK: PLACE IT INTO THE DEFAULT DRIVE, AND TYPE INITIALIZE. PROCESS TAKES ABOUT TWO MINUTES.

OPTIONAL PARAMETERS:

```
INITIALIZE  ["NEW VOL LABEL" [, ":MSUS"      [,DIRECTORY  [,INTER- ]]]]
                                ".OLD VOL"      SIZE        LEAVE
                                LABEL                       FACTOR
```

EXAMPLE: EXECUTING INITIALIZE "DRIVE1",":D710" WILL INITIALIZE THE DISK AT 710 EVENTHOUGH THE DEFAULT DRIVE MAY BE 700. THE DISK MAY BE ACCESSED USING THE VOLUME LABEL .DRIVE1.

NOTES:

HARDWARE: THE CONTROLLER

MASS STORAGE CONTINUED

DISK DIRECTORY

A LISTING OF THE FILES ON A DISK IS CALLED A DIRECTORY OR CATALOG.

ACCESS TO THE DIRECTORY (LIST OF FILES ON THE DISK) IS OBTAINED VIA THE **CAT** STATEMENT.

FORMAT

```
CAT  [":MSUS"         ]
     [".VOLUME LABEL" ]
```

NOTE: MSUS - MASS STORAGE UNIT SPECIFIER i.e. D700

EXAMPLE: ONLY CAT NEED BE USED IF THE DISK BEING ACCESSED IS AT THE DEFAULT LOCATION.

IF THE DEFAULT IS 700 AND A DIRECTORY LISTING OF 710 IS NEEDED, THEN ENTER **CAT ":D710"**.

RESULT:

```
CAT ":D700"
[ Volume ]:
Name          Type  Bytes   Recs
LAB1-1        PROG    256     10
LAB1-1a       PROG    256     11
LAB2-1        PROG    256     19
```

NAME - NAME ASSIGNED TO FILE

TYPE - DATA
PROG - PROGRAM
BPGM - BINARY PROGRAM
NULL - EMPTY FILE, CREATED WHEN A FILE IS PURGED

MORE DETAILS IN SECTIONS 3 & 4

IN ORDER TO PRINT A COPY OF THE DIRECTORY LISTING ENTER THE PRINT ALL STATEMENT PRIOR TO CAT. TO DISABLE PRINT ALL ENTER NORMAL.

NOTES:

MASS STORAGE CONTINUED

STORING & LOADING PROGRAMS

FORMAT

```
STORE "FILE SPECIFIER" WILL STORE IN THE DEFAULT DRIVE
```

TO SPECIFY A DISK DRIVE OTHER THAN THE DEFAULT DRIVE USE:

```
STORE "FILE SPECIFIER:MSUS"
                     .VOL LABEL
```

EXAMPLES:

STORE "LAB1-1"   STORES PROGRAM FILE LAB1-1 AT DEFAULT DRIVE.

STORE "LAB1-1:D710"   STORES TO DRIVE AT 710, ASSUMING DEFAULT IS 700.

LIMIT FILE NAME TO 10 CHARACTERS, 6 WILL BE DISPLAYED WHEN A DIRECTORY IS DISPLAYED.

* BE CAREFUL NOT TO STORE A PROGRAM WITH A FILE SPECIFIER THAT IS BEING CURRENTLY USED AT THE SPECIFIED DRIVE. THE NEW FILE WILL REPLACE THE OLD. INTENTIONALLY DO THIS WHEN UPDATING A PROGRAM AND NO COPY OF THE ORIGIONAL IS DESIRED.

LOADING PROGRAMS

FORMAT

```
LOAD "FILE SPECIFIER"   WILL LOAD FROM THE DEFAULT DRIVE
```

TO SPECIFY A DISK DRIVE OTHER THAN THE DEFAULT DRIVE USE:

```
LOAD "FILE SPECIFIER:MSUS"
                    .VOL LABEL
```

AUTOSTART PROGRAMS

USING THE FILE NAME **Autost** WILL ALLOW A PROGRAM TO AUTOMATICALLY BE LOADED AND RUN AT POWER UP, IF THE FOLLOWING CONDITIONS ARE MET:

- PROGRAM IS ON THE DEFAULT DRIVE (LOWEST ADDRESS)
- DISK DRIVE IS POWERED ON PRIOR TO THE COMPUTER. AT POWER UP THE COMPUTER SEARCHES THE DIRECTORY FOR Autost.

NOTES:

MASS STORAGE CONTINUED

OTHER FILE OPERATIONS

COPYING FILES

FORMAT

```
COPY "SOURCE FILE SPECIFIER" TO "DESTINATION FILE SPECIFIER"
```

EXAMPLE:

```
COPY "LAB1-1:D700" TO "LAB1-1:D712"
```

THE SAME FILE NAME IS KEPT.

```
COPY "LAB1-1:D712" TO "LAB1-1c"
```

A DUPLICATE WITH A NEW FILE NAME WAS MADE.

COPYING A DISK

FORMAT

```
COPY ":SOURCE MSUS" TO ":DESTINATION MSUS"
     ".SOURCE VOL LABEL" TO ".DESTINATION VOL LABEL"
```

EXAMPLE:

```
COPY ":D700" TO ":D710"
```

RENAMING FILES

FORMAT

```
RENAME "OLD FILE SPECIFIER" TO "NEW FILE NAME"
```

EXAMPLE:

```
RENAME "LAB1-1:D700" TO "LAB1A"
```

PURGING FILES

PREVENTS FURTHER DIRECTORY ACCESS TO FILE.

FORMAT

```
PURGE "FILE SPECIFIER"

PURGE "FILE SPECIFIER" .0
```

EXAMPLES:

`PURGE "LAB1-1:D700"` PURGES FILE FROM DISK, IS NULL FILE

`PURGE "LAB1-1".0` PURGES FILE AND THOSE FOLLOWING FROM DIRECTORY, NO NULL FILES CREATED.

NOTES:

HARDWARE: THE CONTROLLER

MASS STORAGE CONTINUED

PACKING A DISK - REMOVES NULL FILES, CREATES MORE SPACE.

FORMAT

```
PACK [":MSUS"     ]
     [".VOL LABEL"]
```

SECURING FILES - THIS ENABLES FILES TO BE SECURED AGAINST UNAUTHORIZED ACCESS.

FORMAT

```
SECURE "FILE SPECIFIER","SECURITY CODE",SECURITY TYPE
```

SECURITY CODE - A QUOTED STRING OF WHICH THE FIRST TWO LETTERS ARE USED.

SECURITY TYPE

0 - PROG - PREVENTS LIST, PLIST, FILE NAME REMAINS IN DIRECTORY, CAN BE LOADED AND RUN.
1 - PROG - SAME AS 0, PREVENTS DUPLICATION. CANNOT STORE IN ANOTHER FILE ONCE LOADED.
2 - PROG, DATA - PREVENTS FILE FROM BEING OVERWRITTEN, CAN BE DUPLICATED.
3 - PROG, DATA - REMOVES FILE NAME FROM DIRECTORY, CAN BE ACCESSED BY KNOWING ASSIGNED NAME.

EXAMPLES:

```
SECURE "LAB006:D712","JBOND",0   SECURITY CODE IS JB

SECURE "LAB007:D712","JBOND",3
```

DIRECTORY LISTING

```
{ Volume }: CCTS
Name          Type    Bytes    Recs
LAB006        PROG    256      4
              PROG    256      10
```

REMOVING FILE SECURITY

- CANCELS SECURITY FOR A FILE

FORMAT

```
UNSECURE "FILE SPECIFIER","SECURITY CODE",SECURITY TYPE
```

EXAMPLE:

```
UNSECURE "LAB007:D712","JB",3
```

```
10 !  LAB 2-1 CONTROLLER FAMILIARIZATION
20 !
30 ! THE PURPOSE OF THIS LAB IS TO FAMILIARIZE THE STUDENT WITH CONTROLLER
      FEATURES, OPERATION, AND EDITING CAPABILITIES.
40 !
50 ! KEYBOARD FAMILIARIZATION
60 !    CALCULATOR MODE: PERFORM THE FOLLOWING MATHEMATICAL EXPRESSION
                         Xc=1/2 FC, F=123.27KHz,C=0.0047uF, Xc=________
70 !      WRITE YOUR EXPRESSION AS ENTERED INTO THE COMPUTER:
          _____________________________________________________________
80 !    PROGRAM MODE: DURING THIS PORTION OF THE LAB YOU WILL BE USING THE
          EDITING FEATURES TO MODIFY THE PROGRAM BELOW.
90 !      USE THE FOLLOWING KEYS TO EDIT THE PROGRAM TO THE LEFT SO IT
          MATCHES THE ONE TO THE RIGHT: BACK SPACE, -LINE, -CHAR, I/R,
100 !     AND CURSOR KEYS. REMOVE THE FAR LEFT ! AS EACH LINE IS CORRECTED
110 !     PRIOR TO END LINE, STARTING FROM LINE 180!!!!!!!!!!!!!!
120 !
130 !   THIS PROGRAM INSTRUCTS THE           130 !     THIS PROGRAM INSTRUCTS THE
140 !   dm TO TAKEE10 READINGZ,              140 !     DMM TO TAKE 100 READINGS,
150 !   STRE THEM INAN ARAY, AND.....        150 !     STORE THEM IN AN ARRAY, AND
160 !   (WITH VERY FEW STEPS)                160
170 !   AAAVERAGEE THEM?                     170 !     AVERAGE THEM.
180 !                                        180 !
190 ! GO SUB SETUP!                          190 GOSUB SETUP !
200 !                                        200 GOSUB KEYTEST !
210 ! GOSUB AVERAGE !                        210 GOSUB AVG !
220 ! GOSUB ERROR !                          220 !
230 ! GOSUB FINAL VAL !                      230 GOSUB FINAL_VAL !
240 ! GOSUB THE__END !                       240 GOTO THE_END !
250 ! SETUP; ! DMM INITIALIZATION            250 SETUP: ! DMM INITIALIZATION
260 ! RESET 07 ! RESETS INTERFACE #7         260 RESET 7 ! RESETS INTERFACE #7
270 ! CLEAR 07! DEVICES TO DEFAULT           270 CLEAR 7 ! DEVICES TO DEFAULT
280 ! REMOTE732 ! TO BUS CONTROL             280 REMOTE 723 ! TO BUS CONTROL
290 ! LOCALLOCKOUT7 ! FRONT DISABLE          290 LOCAL LOCKOUT 7 ! FRONT DISABLE
300 ! OUTPUT 723;"HO" ! DC, NO TRIG          300 OUTPUT 723;"HO" ! DC, NO TRIG
310 ! RETURN, ! TO MAIN PRGM                 310 RETURN ! TO MAIN PRGM
320 ! KEY TEST: ! K5 TO START                320 KEYTEST: ! K5 TO START
330 ! ON KEY#5,'RUN TEST' GOTO START !       330 ON KEY#5,"RUN TEST" GOTO START !
340 ! KEY LABEL: ! DISPLAYS LABEL            340 KEY LABEL ! DISPLAYS LABEL
350 ! FLASH; DISP PRESS K5 TO START !        350 FLASH: DISP "PRESS K5 TO START" !
360 ! WAIT 3000 ! 3 SEC DELAY                360 WAIT 3000 ! 3 SEC DELAY
370 ! CLEER ! BLANKS SCREEN                  370 CLEAR ! BLANKS SCREEN
380 ! KEYLABEL ! DISPLAYS LABEL              380 KEY LABEL ! DISPLAYS LABEL
390 ! WAIT 1000 ! DELAY .1 SEC               390 WAIT 100 ! DELAY .1 SEC
400 ! GO TO FLASH ! LOOP                     400 GOTO FLASH ! LOOP
410 ! START: DISP"TEST STARTED ! MSG         410 START: DISP "TEST STARTED" ! MSG
420 ! RETURM ! TO MAIN PRGM                  420 RETURN ! TO MAIN PRGM
430 ! AVERAGE: ! TAKE READINGS & AVG         430 AVG: ! TAKE READINGS & AVG
440 ! DIM VOLTS (100)(13) ! SETUP ARRAY      440 DIM VOLT$ (100)[13] ! SETUP ARRAY
450 ! READ# = 0 ! SET VARIABLE TO 0          450 READING# = 0 ! SET VARABLE TO 0
460 ! FOR N=1 TO 100 ! SETUP LOOP            460 FOR N=1 TO 100 !SETUP LOOP
470 ! OUTPUT 723, "T3" ' TRIG DMM            470 OUTPUT 723; "T3" ! TRIG DMM
480 ! ENTER 723; VOLT$(READING) ! INPUT      480 ENTER 723; VOLT$ (READING#) ! INPUT

490 ! NEXT M ! LOOP BACK OR NOT              490 NEXT N ! LOOP BACK OR NOT
500 ! SUM$-0 ! SET VARIABLE TO 0             500 SUM$=0 ! SET VARIABLE TO 0
510 ! FOR N=1 TO 99 ! SETUP LOOP             510 FOR M=0 TO 99 ! SETUP LOOP
520 ! SUM$ = SUM$ + VOLT$(N)[1,11] ! ADD 520 SUM$ = SUM$ + VOLT$(M)[1,11] ! ADD
530 ! DISP "AVERAGE OF READINGS =  !         530 DISP "AVERAGE OF READINGS = " !
```

```
540 ! RETERN ! TO MAIN PRGM                 540 RETURN ! TO MAIN PRGM
550 ! FINAL-VAL: ! PRINT RESULT            550 FINAL_VAL: ! PRINT RESULT
560 ! DISP 'AVG VALUE IS '; SUM$ !         560 DISP "AVG VALUE IS "; SUM$ !
570 ! PRINTERS IS 705 ! SETUP PRINTER     570 PRINTER IS 705 ! SETUP PRINTER
580 ! PRINT "AVG VALUE IS", SUM$ !         580 PRINT "AVG VALUE IS "; SUM$ !
590 ! PRINT USENG "/////" ! CrLf           590 PRINT USING "/////" ! CrLf
600 ! RETURN. ! TO MAIN PRGM               600 RETURN ! TO MAIN PRGM
610 ! THE END: ! PROGRAM END               610 THE_END: PROGRAM END
620 ! LOCAL: ! ENABLE FRONT PANEL          620 LOCAL ! ENABLE FRONT PANEL
630 ! THE END ! ALL DONE                   630 END ! ALL DONE
640 !
650 !
660 !
670 ! BE SURE TO STORE YOUR CORRECTED VERSION BEFORE YOU ARE DONE!!!!!!
680 ! STORE IT UNDER THE FILE NAME LB2-1X
690 ! REMEMBER THE SEQUENCE FOR STOREING IS:
700 ! MASS STORAGE IS ":D700"
710 ! PLACE YOUR DISK INTO THE DRIVE
720 ! ENTER THE FOLLOWING STATEMENT:    STORE "LB2-1X"  THEN PRESS END LINE
730 ! YOU WILL BE LOADING THIS VERSION WHEN YOU ANSWER THE QUESTIONS THAT FOLLOW
```

HARDWARE: THE CONTROLLER

LAB EXERCISES CONTINUED

LAB2-1 ANALYSIS

1. WILL THE COMPUTER REJECT A PROGRAM STATEMENT (SYNTAX ERROR) IF THE COMMENT ON THE LEFT LISTING DOES NOT MATCH THE ONE TO THE RIGHT?

2. WOULD THE PROGRAM RUN IF LINE 200 WERE NOT CORRECTED? EXPLAIN

3. IN ORDER TO MAKE THE PROGRAM MORE READABLE SEVERAL BLANK LINES SHOULD BE USED TO SEPERATE SUBROUTINES i.e. ADD THE FOLLOWING TO CREATE BLANK LINES BETWEEN THE MAIN PROGRAM AND THE SUBROUTINE LABELED SETUP. 245 !, 246 !, 247 !
REPEAT THIS PROCESS BETWEEN ALL SUBROUTINES. STORE THE PROGRAM WITH ALL MODIFICATIONS AND CORRECTIONS AS LB2-1X. THIS VERSION WILL BE USED DURING SECTION 5.

4. WILL LINE 230 RUN WITH THE STATEMENT LABEL AS SHOWN IN THE LEFT LISTING? IF NOT WHY?

5. WHAT IS WRONG IN LINE 260? HINT: WHAT IS THE DIFFERENCE BETWEEN O7 AND 07?

6. WILL LINE 280 FUNCTION PROPERLY WITHOUT THE SPACE BETWEEN REMOTE AND 723? IF SO, WHY INCLUDE IT?

LAB EXERCISES CONTINUED

LAB2-1 ANALYSIS

7. WHAT IS THE FUNCTION OF THE ! THAT IS USED THROUGHOUT THE PROGRAM (FOLLOWING EALH STATEMENT)? WHAT ELSE WOULD PERFORM THE SAME FUNCTION?

8. WILL LINE 400 WORK WITH ITS ERROR? BE CAREFUL THIS IS A COMMON MISTAKE!!

9. WHAT IS THE ERROR IN LINE 430? STATEMENT LABELS MUST BE CONSISTANT!!

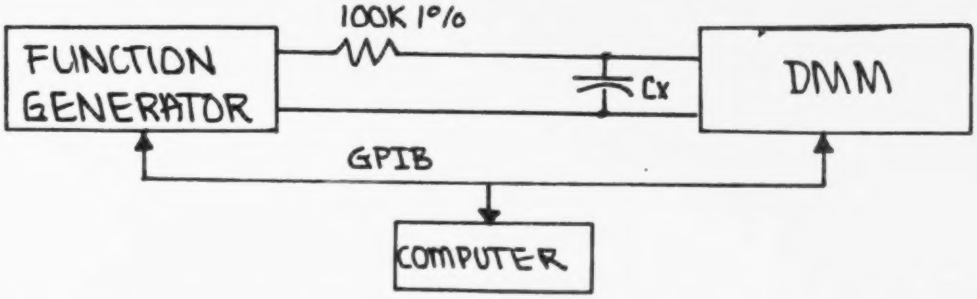
10. WHAT IS THE ERROR IN LINE 470? WHAT OTHER PUNCTUATION MARK MIGHT YOU USE IN ERROR AT THIS POINT?

11. WHAT IS THE ERROR IN LINE 490? THIS TYPE OF PROBLEM IS EASY TO CREATE ESPECIALLY IN MULTIPLE NESTED LOOPS!

12. WHAT IS THE PROBLEM IN LINE 550? WHEN HAND WRITING A PROGRAM BE CAREFUL TO DELINEATE BETWEEN THE DASH AND THE UNDERSCORE CHARACTER!

COMPUTER CONTROLLED TEST SYSTEMS

LAB PROJECT 1: CAPACITANCE MEASUREMENT



1. DESIGN A PROGRAM TO DETERMINE THE CAPACITANCE VALUE OF CAPACITOR Cx FROM TRANSFER FUNCTION MEASUREMENTS (Vo/Vi=0.707 WHEN F=1/(2 Pi RC)).

2. THE CIRCUIT SHOULD FUNCTION OVER THE FOLLOWING RANGE OF Cx: 1000pf <= Cx <= 2uf.

3. THE PROGRAM SHOULD BE WRITTEN IN A USER FRIENDLY FASHION i.e. THE OPERATOR SHOULD BE PROMPTED TO INSTALL THE CAPACITOR AFTER THE OPEN CIRCUIT VOLTAGE HAS BEEN READ.

4. BE SURE TO ALLOW FOR SETTLING TIME BETWEEN THE FUNCTION GENERATOR OUTPUT CHANGES AND DMM READINGS.

5. CHECK YOUR PROGRAM BY USING CAPACITORS AT EACH END, AND THE MIDDLE OF THE MEASUREMENT RANGE. VERIFY THE RESULTS BY CHECKING THE CAPACITORS WITH A CAPACITANCE METER.

6. PREPARE A REPORT WHICH CONTAINS THE FOLLOWING INFORMATION:

   - DETAILS OF YOUR MEASUREMENT METHOD
   - PROGRAM FLOW CHART
   - PROGRAM LISTING
   - ACCURACY OF YOUR MEASUREMENT METHOD

7. THIS PROJECT MUST BE COMPLETED IN ORDER TO RECEIVE A C GRADE IN THIS CLASS.

8. SECURE THIS PROGRAM ON YOUR STORAGE MEDIUM TO PREVENT UNAUTHORIZED COPYING. USE LABPR1 FOR FILE SPECIFIER, USE THE LAST TWO DIGITS OF YOUR STUDENT NUMBER FOR THE SECURITY CODE, USE SECURITY TYPE 3!!

HARDWARE: INSTRUMENT CHARACTERISTICS

PRELUDE - A VAST QUANTITY OF INSTRUMENTS ARE AVAILABLE WHICH SUPPORT SOME LEVEL OF GPIB COMPATABILITY. IN EACH TYPE CATAGORY i.e. DMM, COUNTER, ETC., THERE EXISTS A MULTITUDE OF MAKES AND MODELS. EACH WITH ITS OWN CAPABILITIES AND GPIB COMPATABILITY.

THE FIRST STEP IN SETTING UP A GPIB BASED TEST SYSTEM IS TO DEFINE WHAT IS THE PURPOSE OF THE SYSTEM THAT IS BEING CONFIGURED. * * IN ORDER TO DEVELOP AN ACCURATE DEFINITION, EVALUATE THE FOLLOWING POINTS.

1. WHAT ENVIRONMENT IS THE TEST SYSTEM TO BE USED IN?
   A. PRODUCTION TEST - SPEED IS IMPORTANT
   B. RESEARCH - INCREASED ACCURACY & EASE OF USE

2. SHOULD THE TEST SYSTEM GENERATE STIMULI (TEST SIGNALS)?
   A. IF THE SIGNALS MUST BE CHANGED AND ROUTED TO DIFFERENT TEST POINTS WITHOUT OPERATOR INTERVENTION REQUIRES A PROGRAMMABLE SOURCE AND A PROGRAMMABLE MATRIX SWITCH.
   B. A SOURCE THAT IS JUST TURNED ON AND OFF NEED NOT BE PROGRAMMABLE.

3. WHAT TYPE OF DATA IS TO BE ACQUIRED?
   A. SIMPLE DC OR AC MEASUREMENTS
   B. WAVEFORM DIGITIZATION
   C. NUMBER OF INPUT CHANNELS REQUIRED
   D. DATA ACQUISITION BANDWIDTH REQUIRED
   E. DATA FORMATS GENERATED
   F. CONTROLLER MEMORY CAPACITY

4. DOES DATA REQUIRE REAL TIME PROCESSING?
   A. CAN INSTRUMENTS PREPROCESS DATA i.e. AVERAGE READINGS, ETC.?
   B. CAN DATA BE PROCESSED IN ALLOTED TIME?
   C. AS THE NUMBER OF ACTIVE INSTRUMENTS ON THE BUS INCREASES, THE CONTROLLERS ABILITY TO PROCESS DATA IN REAL TIME DIMINISHES.

5. IS DATA LOGGING REQUIRED?
   A. IN HIGH SPEED DATA ACQUISITION SYSTEMS, THE DATA IS OFTEN DIRECTLY TRANSFERRED TO A MASS STORAGE DEVICE AND PROCESSING OCCURS AT A LATER POINT IN TIME.

NOTES:

HARDWARE: INSTRUMENT CHARACTERISTICS

SELECTION OF INSTRUMENTATION

GENERATE A LIST OF REQUIRED INSTRUMENTS BASED ON SYSTEM DEFINITION REQUIREMENTS. SOME INSTRUMENTS MAY BE REPLACED BY HAVING THE CONTROLLER COMPUTE THE DESIRED PARAMETERS. EXAMPLE: AN OPERATOR MAY ENTER THE PEAK VOLTAGE VALUE OFF AN OSCILLOSCOPE, AND THE CONTROLLER THEN COMPUTES THE RMS VALUE. THIS WOULD ALEVIATE THE NEED FOR A RMS READING DMM. ALTHOUGH THE SYSTEM WOULD BE SLOWER AND MORE ERROR PRONE WITH OPERATOR ENTRY.

THE CONTROLLER MUST PROVIDE COMPLETE GPIB COMPATABILITY. THE SOFTWARE OR FIRMWARE OPERATING SYSTEM MUST:

1. SEND DEVICE DEPENDENT AND INTERFACE MESSAGES i.e. CLEAR 723, CLEAR 7.
2. HANDLE DEVICE DEPENDENT DATA i.e. SET UP INSTRUMENTS, READ THEIR OUTPUT.
3. HANDLE SERVICE REQUESTS (SRQ)
4. UTILIZE HIGH LEVEL SIMPLE COMMANDS TO KEEP APPLICATION PROGRAMMING AS SIMPLE AS POSSIBLE.

NOTES:

HARDWARE: INSTRUMENT CHARACTERISTICS

SELECTION OF INSTRUMENTATION

THE LEVEL OF INSTRUMENT PROGRAMABILITY IS DETERMINED BY SYSTEM NEEDS. KEEP IN MIND THAT OFTEN LOCAL (FRONT PANEL) AND REMOTE (PROGRAMMABLE) CAPABILITIES DIFFER. THIS MEANS THAT ON SOME INSTRUMENTS SOME OF THE FRONT PANEL CAPABILITIES ARE NOT PROGRAMABLE OVER THE BUS. ON OTHER INSTRUMENTS, CAPABILITIES ARE AVAILABLE OVER THE BUS AND NOT VIA THE FRONT PANEL.

THREE LEVELS OF GPIB COMPATABILITY

1. GPIB IS ONLY USED FOR DATA TRANSFER i.e. DISK DRIVE

2. FUNCTIONS ARE PARTIALLY/FULLY PROGRAMABLE - HP 3478A DMM, FLUKE 8840A DMM

3. INSTRUMENTS THAT STORES AND EXECUTES PROGRAMS i.e. PREPROCESSES DATA - TEK 7854 PROGRAMMABLE WAVEFORM PROCESSING SCOPE. FOR FIXED FUNCTION, FIXED RANGE, AND DATA GATHERING, THESE CAPABILITIES ARE NOT NEEDED. FOR AUTOMATIC TESTING THEY ARE. ON BOARD PROCESSING SPEEDS UP PROCESSOR THROUGHPUT.

THE ADVERTISING SLOGANS "GPIB COMPATABLE" OR "GPIB PROGRAMABLE" DO NOT INDICATE THE LEVEL OF COMPATABILITY. READ TECHNICAL LITERATURE CAREFULLY. MOST MANUFACTURERS WILL "TOOT THEIR OWN HORN" IN DESCRIBING THEIR PRODUCTS CAPABILITIES. ALSO, THEY ARE UNLIKELY TO INDICATE FEATURES THEY DO NOT OFFER.

NOTES:

HARDWARE: INSTRUMENT CHARACTERISTICS

SELECTION OF INSTRUMENTATION

MEASUREMENT SPEED IS DEPENDENT ON MANY FACTORS

1. SET UP TIME - PUTTING THE INSTRUMENT INTO THE DESIRED MODE. THE TIME REQUIRED FOR THE INSTRUMENT TO RESPOND TO THE COMMAND.

2. ACQUISITION

   A. TRIGGER DELAY - TIME DELAY TO TRIGGER, ONCE TOLD TO TRIGGER.

   REPRINTED WITH PERMISSION, COURTESY OF TEKTRONIX, INC.

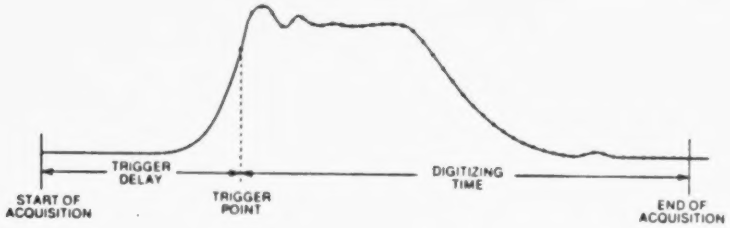


Fig B—Data acquisition is composed of two parts. *The first part, trigger delay, is the time between when the instrument is ready to begin acquisition and when the trigger occurs. The second part, digitizing time, is the time required to actually capture the data.*

   B. INSTRUMENT PROCESSING TIME - DMM A/D CONVERSION TIME FOR EXAMPLE.

3. DATA TRANSFER TIME - RELATED TO NUMBER OF BYTES TO BE TRANSFERRED AND BYTE TRANSFER RATE.

IF SEVERAL DEVICES HAVE BEEN SET UP TO RESPOND TO A COMMON TRIGGER i.e. TWO DMMs GATHERING DATA SIMULTANEOUSLY. IF THE DMM WHICH TRANSFERS DATA SLOWEST RESPONDS FIRST, THE CONTROLLER WILL BE "TIED UP" UNTIL THE SLOWEST DMM IS COMPLETE, BEFORE IT CAN RESPOND TO THE DMM HAVING THE FASTER TRANSFER RATE.

4. DATA PROCESSING TIME - NUMERICAL ANALYSIS IS OFTEN DONE BY THE CONTROLLER. EXAMPLES: LINEARIZATION OF THERMOCOUPLE VOLTAGE READINGS, STATISTICAL ANALYSIS, ETC.

5. HUMAN INTERACTION - PROMPTING OPERATOR TO CHANGE A NON-PROGRAMABLE FUNCTION, ENTER TEST PARAMETERS, CHANGE UNIT UNDER TEST (UUT). THE HUMAN INTERACTION FACTOR SHOULD BE MINIMIZED, AUTOMATE WHENEVER POSSIBLE.

NOTES:

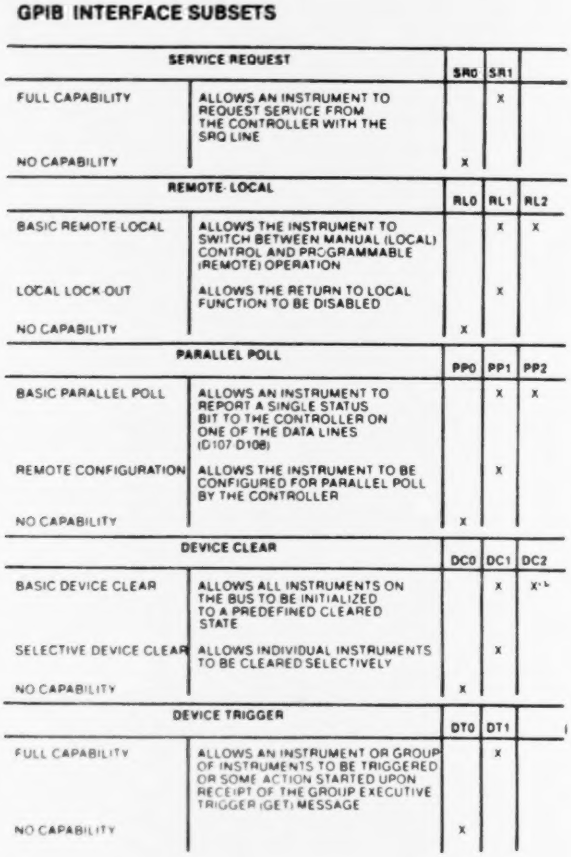
HARDWARE: INSTRUMENT CHARACTERISTICS

SELECTION OF INSTRUMENTATION

DETERMINE WHICH INTERFACE FUNCTIONS ARE IMPLEMENTED

1. IEEE 488 DEFINES THE FOLLOWING INTERFACE FUNCTIONS (THE CHOICE OF WHICH FUNCTIONS TO IMPLEMENT IN AN INSTRUMENT IS LEFT UP TO THE MANUFACTURER, THUS NOT ALL FUNCTIONS ARE IMPLEMENTED ON ALL INSTRUMENTS.

REPRINTED WITH PERMISSION, COURTESY TEKTRONIX, INC.



GPIB INTERFACE SUBSETS

SERVICE REQUEST		SR0	SR1	
FULL CAPABILITY	ALLOWS AN INSTRUMENT TO REQUEST SERVICE FROM THE CONTROLLER WITH THE SRQ LINE		X	
NO CAPABILITY		X		
REMOTE-LOCAL		RL0	RL1	RL2
BASIC REMOTE-LOCAL	ALLOWS THE INSTRUMENT TO SWITCH BETWEEN MANUAL (LOCAL) CONTROL AND PROGRAMMABLE (REMOTE) OPERATION		X	X
LOCAL LOCK-OUT	ALLOWS THE RETURN TO LOCAL FUNCTION TO BE DISABLED		X	
NO CAPABILITY		X		
PARALLEL POLL		PP0	PP1	PP2
BASIC PARALLEL POLL	ALLOWS AN INSTRUMENT TO REPORT A SINGLE STATUS BIT TO THE CONTROLLER ON ONE OF THE DATA LINES (D107-D108)		X	X
REMOTE CONFIGURATION	ALLOWS THE INSTRUMENT TO BE CONFIGURED FOR PARALLEL POLL BY THE CONTROLLER		X	
NO CAPABILITY		X		
DEVICE CLEAR		DC0	DC1	DC2
BASIC DEVICE CLEAR	ALLOWS ALL INSTRUMENTS ON THE BUS TO BE INITIALIZED TO A PREDEFINED CLEARED STATE		X	X
SELECTIVE DEVICE CLEAR	ALLOWS INDIVIDUAL INSTRUMENTS TO BE CLEARED SELECTIVELY		X	
NO CAPABILITY		X		
DEVICE TRIGGER		DT0	DT1	
FULL CAPABILITY	ALLOWS AN INSTRUMENT OR GROUP OF INSTRUMENTS TO BE TRIGGERED OR SOME ACTION STARTED UPON RECEIPT OF THE GROUP EXECUTIVE TRIGGER (GET) MESSAGE		X	
NO CAPABILITY		X		

GPIB INTERFACE SUBSETS CHART CONTINUED ON Pg. 3-6

EXAMPLE: THE HP 3478A DMM IMPLEMENTS THE FOLLOWING INTERFACE FUNCTIONS - SH1, AH1, T5, L4, LE0, TE0, SR1, RL1, PP0, DC1, DT1, C0.

NOTES:

SELECTION OF INSTRUMENTATION

DETERMINE WHICH INTERFACE FUNCTIONS ARE IMPLEMENTED

REPRINTED WITH PERMISSION, COURTESY TEKTRONIX, INC.

SOURCE HANDSHAKE		SH0	SH1
FULL CAPABILITY	ALLOWS A DEVICE TO GENERATE THE HANDSHAKE CYCLE FOR TRANSMITTING DATA		X
NO CAPABILITY		X	

ACCEPTOR HANDSHAKE		AH0	AH1
FULL CAPABILITY	ALLOWS A DEVICE TO GENERATE THE HANDSHAKE FOR RECEIVING DATA		X
NO CAPABILITY		X	

TALKER (EXTENDED TALKER)*		T0 (TE0)	T1 (TE1)	T2 (TE2)	T3 (TE3)	T4 (TE4)	T5 (TE5)	T6 (TE6)	T7 (TE7)	T8 (TE8)
BASIC TALKER (BASIC EXTENDED TALKER)	ALLOWS AN INSTRUMENT TO TRANSMIT DATA		X	X	X	X	X	X	X	X
TALK ONLY MODE	ALLOWS AN INSTRUMENT TO TRANSMIT DATA WITHOUT A CONTROLLER ON THE BUS		X		X		X		X	
UNADDRESSED IF MY LISTEN ADRESS (MLA)	PREVENTS AN INSTRUMENT FROM BEING A TALKER AND A LISTENER AT THE SAME TIME						X	X	X	X
SERIAL POLL	ALLOWS AN INSTRUMENT TO SEND A STATUS BYTE IN RESPONSE TO A SERIAL POLL		X	X			X	X		
NO CAPABILITY		X								

LISTENER (EXTENDED LISTENER)*		L0 (LE0)	L1 (LE1)	L2 (LE2)	L3 (LE3)	L4 (LE4)
BASIC LISTENER (BASIC EXTENDED LISTENER)	ALLOWS AN INSTRUMENT TO RECEIVE DATA		X	X	X	X
LISTEN ONLY MODE	ALLOWS AN INSTRUMENT TO RECEIVE DATA WITH A CONTROLLER ON THE BUS		X		X	
UNADDRESS IF MY TALK ADDRESS (MTA)	PREVENTS AN INSTRUMENT FROM BEING A TALKER AND A LISTENER AT THE SAME TIME				X	X
NO CAPABILITY		X				

*EXTENDED TALKERS AND LISTENERS USE SECONDARY ADDRESSES OTHER TALKERS AND LISTENERS DO NOT

SH1 - ALLOWS THE DMM TO INITIATE THE COMMUNICATION (HANDSHAKE) OF INFORMATION TO THE CONTROLLER i.e. SEND A READING WHEN IT IS READY.

AH1 - ALLOWS THE DMM TO INITIATE THE COMMUNICATION (GENERATE THE HANDSHAKE RESPONSE TO A SOURCE HANDSHAKE) WITH A DEVICE WHICH IS SENDING IT DATA.

T5 - TALKER

BASIC TALKER - INSTRUMENT CAN SEND DATA

SERIAL POLL - SENDS STATUS BYTE IN RESPONSE TO A SERIAL POLL. THIS IS NEEDED IN ORDER FOR THE CONTROLLER TO RESPOND TO A SRQ.

TALK ONLY - THIS ALLOWS THE DMM TO SEND DATA DIRECTLY TO A LISTENING DEVICE i.e. A PRINTER, WITHOUT THE USE OF A CONTROLLER. THIS ALLOWS DATA LOGGING SYSTEMS TO BE SET UP WITH ONLY A DMM AND A PRINTER.

HARDWARE: INSTRUMENT CHARACTERISTICS

SELECTION OF INSTRUMENTATION

DETERMINE WHICH INTERFACE FUNCTIONS ARE IMPLEMENTED

UNADDRESSED IF MY LISTEN ADDRESS (MLA) - IF AN INSTRUMENT IS PUT IN THE LISTEN MODE, IT IS UNADDRESSED TO TALK. THUS IT CANNOT TALK AND LISTEN AT THE SAME TIME i.e. IT CANNOT LISTEN TO ITSELF TALK, THE CONTROLLER IS THE TALKER.

L4 - LISTENER

BASIC LISTENER - THE INSTRUMENT HAS THE ABILITY TO RECEIVE DATA

UNADDRESSED IF MY TALK ADDRESS (MTA) - IF PUT INTO THE TALK MODE, IT IS UNADDRESSED TO LISTEN. THIS PREVENTS THE INSTRUMENT FROM LISTENING AND TALKING AT THE SAME TIME.

LE0 - EXTENDED LISTENER, NO CAPABILITY

TE0 - EXTENDED TALKER, NO CAPABILITY. THE EXTENDED ADDRESSING CAPABILITIES GIVE AN INSTRUMENT THE ABILITY TO HAVE A LOWER LEVEL OR FUNCTION ACCESSED WITH A SECONDARY COMMAND. THESE COULD ACCESS A PARTICULAR REGISTER OR CARD WITHIN THE INSTRUMENT.

SR1 - THIS ALLOWS THE INSTRUMENT TO REQUEST SERVICE FROM THE CONTROLLER USING THE SRQ LINE. THUS THE INSTRUMENT CAN REQUEST SERVICE ASYNCHRONOUS TO THE APPLICATION PROGRAM FLOW.

RL1 - ALLOWS THE CONTROLLER TO SWITCH THE INSTRUMENT BETWEEN LOCAL (FRONT PANEL) AND REMOTE (PROGRAMMABLE) OPERATION. BY PLACING AN INSTRUMENT IN PROGRAM ONLY CONTROL OF ITS FUNCTIONS, THIS WILL PREVENT FRONT PANEL CHANGING OF THE INSTRUMENTS SET UP.

PP0 - PARALLEL POLL, NO CAPABILITY. INSTRUMENTS WITH PARALLEL POLL CAPABILITY WILL SEND THE CONTROLLER A SINGLE STATUS BIT ON ONE OF THE EIGHT DATA LINES (DIO1-DIO8) INDICATING WHICH DEVICE IS REQUESTING SERVICE i.e. SRQ.

WHEN A SRQ IS INITIATED, THE CONTROLLER IS INTERRUPTED (VIA SRQ LINE). IN ORDER FOR THE CONTROLLER TO KNOW WHICH INSTRUMENT ACTIVATED THE SRQ LINE IT MUST PERFORM A PARALLEL POLL. ONCE THE REQUESTING DEVICE IS LOCATED IT MUST BE SERIAL POLLED IN ORDER TO DETERMINE ITS SPECIFIC REQUEST. WHEN SERIAL POLLED THE INSTRUMENT WILL SEND A STATUS BYTE WHOSE BITS, WHEN INTERROGATED, INDICATE TO THE CONTROLLER WHAT NEEDS TO BE DONE. SINCE A SERIAL POLL IS REQUIRED TO DETERMINE THE REASON FOR THE SRQ, AND ONLY A SERIAL POLL WILL RESET THE SRQ LINE, PARALLEL POLLS ARE NOT USED THAT OFTEN. THE REAL ADVANTAGE OF PARALLEL POLLS IS IN LARGE SYSTEMS WHERE SERIAL POLLING THROUGH ALL DEVICES IS TIME CONSUMING.

NOTES:

HARDWARE: INSTRUMENT CHARACTERISTICS

SELECTION OF INSTRUMENTATION

DETERMINE WHICH INTERFACE FUNCTIONS ARE IMPLEMENTED

DC1 - DEVICE CLEAR

BASIC DEVICE CLEAR - ALL INSTRUMENTS ON THE BUS WHICH INCORPORATE DC1 OR DC2 MAY BE SET TO A PREDEFINED STATE, OFTEN THE POWER ON STATE.

EXAMPLE: CLEAR 7 WILL CLEAR ALL DEVICES ON INTERFACE 7.

SELECTIVE DEVICE CLEAR - THE INSTRUMENT WHICH IS ADDRESSED IN THE CLEAR STATEMENT IS THE ONLY INSTRUMENT WHICH IS SET TO THE PREDEFINED STATE.

EXAMPLE: CLEAR 723 WILL CLEAR THE DEVICE AT ADDRESS 723.

DT1 - DEVICE TRIGGER

UPON THE RECEIPT OF A GROUP EXECUTIVE TRIGGER (GET) MESSAGE THE INSTRUMENT OR GROUP OF INSTRUMENTS SELECTED TO PERFORM A SPECIFIED ACTION.

EXAMPLE: TRIGGER 7 WILL TRIGGER ALL DEVICES ON INTERFACE 7
WHICH HAVE REMOTE TRIGGER CAPABILITY.

TRIGGER 723 WILL TRIGGER THE DEVICE AT ADDRESS 723.

TRIGGER 723,713 WILL TRIGGER THE DEVICES AT ADDRESSES 723 and 713.

C0 - CONTROLLER, NO CAPABILITY. DEVICES WITH THIS CAPABILITY CAN ACT AS A CONTROLLER i.e. THEY CAN SEND ADDRESSES, UNIVERSAL COMMANDS, AND ADDRESSED COMMANDS TO OTHER DEVICES ON THE BUS.

NOTES:

HARDWARE: INSTRUMENT CHARACTERISTICS

SELECTION OF INSTRUMENTATION

MESSAGE FORMAT IS NOT STANDARD

IEEE 488 DOES NOT SPECIFY BUS MESSAGE CODING OR SYNTAX.

ASCII IS THE MOST COMMON FORMAT FOR DATA.

THE MOST SIGNIFICANT DIGIT (MSD) IS MOST OFTEN SENT FIRST.

NO STANDARD EXISTS FOR DATA OTHER THAN NUMBERS.

IN MANY CASES THE COMMANDS WHICH THE CONTROLLER OUTPUTS TO SET UP AN INSTRUMENT MAY NOT RESEMBLE THE DATA ITSELF. EXAMPLE: TO SET UP A POWER SUPPLY TO OUTPUT 18.5V THE STRING "2370" IS SENT TO THE POWER SUPPLY. THE 2 PUTS THE SUPPLY INTO ITS SECOND RANGE (50V FS) AND THE REMAINING THREE CHARACTERS FIT INTO THE NUMERATOR OF THE FOLLOWING EXPRESSION (XXX/1000)X50 i.e. (370/1000)X50 = 18.5.

WHEN SELECTING INSTRUMENTS IF AT ALL POSSIBLE SELECT INSTRUMENTS WITH THE SAME OR SIMILAR FORMATS. THIS WILL MAKE CODING (GENERATION OF SOFTWARE) MUCH EASIER. SINCE THE SOFTWARE IS A MAJOR PORTION OF THE TEST SYSTEM (COST AND TIME).

MANUFACTURERS ARE SETTING UP INTERNAL FORMAT AND SYNTAX STANDARDS, SO ALL OF THEIR INSTRUMENTATION USES THE SAME TYPE COMMANDS FOR THE SAME OPERATIONS. PROBLEMS CAN EXIST WHEN USING INSTRUMENTS WHICH WERE DESIGNED PRIOR TO AND AFTER THE STANDARDS WERE SET. EXAMPLE: "F" MAY SET THE FREQUENCY ON ONE INSTRUMENT, BUT ANOTHER MAY REQUIRE "FR".

A MESSAGE TERMINATOR IS USED TO SPECIFY THE END OF A MESSAGE

A CARRAGE RETURN (Cr) AND/OR LINE FEED (Lf) ARE OFTEN USED TO SIGNIFY THE END OF A MESSAGE.

IF AN INSTRUMENT SENDS A CrLf TERMINATOR AND A CONTROLLER JUST USES A Cr, THEN THE Lf WILL BE SENT AS PART OF THE NEXT MESSAGE STRING, THUS CAUSING AN ERROR.

SOME INSTRUMENTS/CONTROLLERS USE THE END OR IDENTIFY LINE (EOI) TO SPECIFY MESSAGE TERMINATION.

OFTEN TERMINATION CAN BE CHANGED VIA SOFTWARE OR DIP SWITCHES.

NOTES:

HARDWARE: INSTRUMENT CHARACTERISTICS

DMM CHARACTERISTICS

THE HP 3478A DMM HAS BEEN CHOSEN AS A REPRESENTATIVE DMM. THE HEWLETT PACKARD PUBLICATION "USING THE 3478A" PRODUCT NOTE 3478A-1 WILL BE REFERENCED IN THIS SECTION. THIS PUBLICATION IS AVAILABLE FREE OF CHARGE FROM YOUR LOCAL HP SALES OFFICE.

WITH THE ADVENT OF THE MICROPROCESSOR INSTRUMENTS ARE BECOMMING SMARTER i.e. THE ABILITY TO PROCESS DATA, AND HAVE INTELLIGENT ALPHA-NUMERIC DISPLAYS. THE 3478A HAS A DISPLAY WHICH CAN DISPLAY READINGS, AND 12 DIGIT MESSAGES TO PROMPT THE OPERATOR.

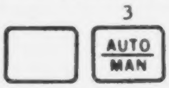
REPRINTED WITH PERMISSION, COURTESY HEWLETT PACKARD CO.

Display

The 3478A's display provides readings with selectable 3½ – 5½ digit resolution. Measurement units (VDC, VAC, KOHM, etc.) are displayed with the reading. Messages may also be displayed with the 12 character alphanumeric liquid crystal display (LCD).

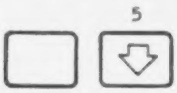
Number of Displayed Digits

Selecting the number of digits displayed also controls the integration time of the 3478A's A/D converter. Thus, the number of digits displayed controls resolution, measurement speed, accuracy, and noise rejection (see "Reading Rates", pg.8).



This puts the 3478A into the 3½ digit display mode. This mode has the fastest reading rate but the lowest resolution and no normal mode rejection. The integration time in this mode is 0.1 power line cycle.

This is the 4½ digit display mode. This mode provides 59 dB of normal mode rejection, with an integration time of 1 power line cycle (16.66 mS at 60 Hz, 20 mS at 50 Hz).



The 5½ digit display mode provides the greatest resolution and noise rejection. In this mode, 10 readings are taken, each with 1 power line cycle integration time, and averaged together. This provides 80 dB of normal mode rejection.

Displayed Messages

The 3478A will display messages indicating the HP-IB address, user errors, self-test failures and calibration status (see 3478A Operator's Manual). User generated messages of up to 12 alphanumeric characters may also be displayed using the "D2" HP-IB command. Examples could include status prompts such as "TEST PASSED" or readings in the desired units such as "257.8 DEG C". To display a message such as "NEXT TEST", simply send the following command to the 3478A: "D2 NEXT TEST".

COMMANDS

N3	Selects 3½ digit display
N4	Selects 4½ digit display
N5	Selects 5½ digit display
D1	Selects normal readings display
D2	User generated message mode. (D2XXXXXXXXXXXX places the message "XXXXXXXXXXXX" on the display
D3	Same as **D2** except that the display is not updated. (Sending only "**D3**" to the 3478A simply turns the display off.)

EXAMPLE: THE STATEMENT OUTPUT 723;"D2CONNECT UUT" WILL DISPLAY THE MESSAGE CONNECT UUT.

NOTES:

DMM CHARACTERISTICS

IN ORDER TO SET UP THE DMM TO TAKE THE DESIRED READING, A COMMAND STRING MUST BE SENT TO IT TO SET UP THE FUNCTION, RANGE, AND TRIGGER.

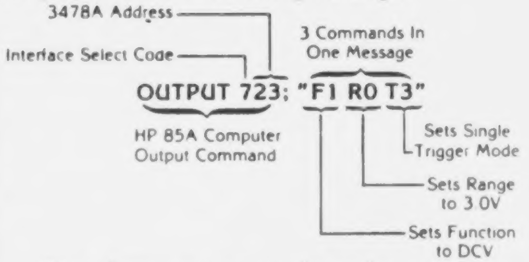
REPRINTED WITH PERMISSION, HEWLETT PACKARD CO.

Sending Commands to the 3478A

You can send commands to the 3478A to select the measurement function, range or resolution; trigger a reading; or interrogate the status of the 3478A. These instructions are sent via the HP-IB command codes listed at the end of this section.

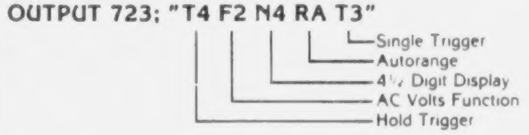
Procedure

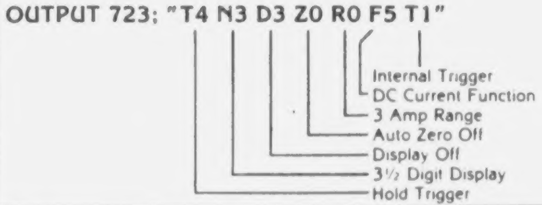
Simply decide what you want the 3478A to do and select the appropriate HP-IB command codes. For example, the command code for dc volts is F1, the code for the 3V range is R0, and single trigger mode is selected by T3. Triggering the 3478A for a single dc volts measurement on the 3V range is accomplished with the following message:


3478A Address
Interface Select Code
3 Commands In One Message
OUTPUT 723; "F1 R0 T3"
HP 85A Computer Output Command
Sets Single Trigger Mode
Sets Range to 3.0V
Sets Function to DCV

More than one command may be sent to the 3478A at a time. Commands are implemented in the order they are received over HP-IB. Thus, in the above message, the 3478A would go to dc volts first, then go to the 3V range, then would trigger once.

Some other examples:


OUTPUT 723; "T4 F2 N4 RA T3"
Single Trigger
Autorange
4½ Digit Display
AC Volts Function
Hold Trigger


OUTPUT 723; "T4 N3 D3 Z0 R0 F5 T1"
Internal Trigger
DC Current Function
3 Amp Range
Auto Zero Off
Display Off
3½ Digit Display
Hold Trigger

Home Commands

The 3478A has a series of 8 preset commands which can provide programming convenience. These commands, called HOME commands, set the 3478A to the predetermined states as shown below.

Home Commands

HP-IB Command	Description
H0	HOME Command. The 3478A is placed into the dc volts function (F1), autorange (RA), trigger hold (T4), 4½ digit display (N4), and auto zero on (Z1). Any result ready to be output to the bus or display is erased. External trigger input is disabled. This command is equal to: "F1T4R−2RAZ1N4". The meter will wait for a trigger command such as T3 or T5.
H1	Measure DC Volts. This function is the same as the H0 command except that a single trigger occurs, and if the 3478A is addressed to talk, will output the reading. This command is equivalent to: "F1R−2RAZ1N4T3".
H2	Measure AC Volts. This command is the same as H1 except that an ac voltage measurement is taken. Equivalent to: "F2R−2RAZ1N4T3"
H3	Measure 2-Wire Ohms. Identical to H1 except takes a 2-wire ohms measurement.
H4	Measure 4-Wire Ohms. Identical to H1 except takes a 4-wire ohms measurement.
H5	Measure DC Current. Identical to H1 except takes a dc current measurement.
H6	Measure AC Current. Identical to H1 except takes an ac current measurement.
H7	Measure Extended Ohms. Identical to H1 except takes an extended ohms measurement.

You can use the HOME commands just as they are or combined with other commands. For example:

OUTPUT 723; "H1"

would set the 3478A to the dc volts function and take one reading. If you want the 3478A to be in the internal trigger mode, you could send:

OUTPUT 723; "H1T1"

In order for "T1" to be in effect, it must follow the "H1" command, otherwise the "H1" command will put "T3" in effect.

NOTES:

HARDWARE: INSTRUMENT CHARACTERISTICS

DMM CHARACTERISTICS

3478A HP-IB PROGRAMMING COMMANDS
REPRINTED WITH PERMISSION, HEWLETT PACKARD CO.

Type	Command	Explanation
Measurement Function	F1	DC Volts function
	F2	AC Volts function
	F3	2-Wire Ohms function
	F4	4-Wire Ohms function
	F5	DC Current function
	F6	AC Current function
	F7	Extended Ohms function
Range	R-2	30 mV dc range
	R-1	300 mV ac or dc range, or the 300 mA ac or dc range
	R0	3V ac or dc range, or 3A ac or dc range
	R1	30V ac or dc range, or the 30 ohm range
	R2	300V dc or ac range, or the 300 ohm range
	R3	3 kΩ range
	R4	30 kΩ range
	R5	300 kΩ range
	R6	3 MΩ range
	R7	30 MΩ range
	RA	Selects Autorange
An invalid range for a given function defaults to the most sensitive range if the range specified is too low, or the least sensitive range if the specified range is too high.		
Display	N3	Selects the 3½ digit display. Fastest reading but little noise rejection. .1 Power Line Cycle integration.
	N4	Selects the 4½ digit display. 1 PLC integration.
	N5	Selects the 5½ digit display. Best resolution and greatest noise rejection. 10 PLC integration.
Trigger	T1	Internal Trigger
	T2	External Trigger. Any reading in progress is aborted and the 3478A waits for an external trigger pulse to occur. Triggering occurs on the negative going edge of the pulse. Readings may also be initiated by an HP-IB GET command.
	T3	Single Trigger. This causes a single measurement to commence. Further readings may be initiated by an HP-IB GET command, but not an external trigger pulse.
	T4	Trigger Hold. Any reading in progress is aborted and the 3478A remains idle. The 3478A will still respond to an HP-IB GET command but not to an external trigger.
	T5	Fast Trigger. Fast Trigger works the same as T3 except that in ac volts, ac current, or the two highest ohms range, the initial settling delay is omitted. If more than one reading is necessary because of autoranging or if the dmm must change ranges or functions to perform the measurement, the delay will occur normally.
Auto Zero	Z0	Auto Zero Off
	Z1	Auto Zero On
Write To Display	D1	Return from D2 or D3 to NORMAL DISPLAY
	D2text	The "D2" command places the message "text" into the 3478A display. The message can contain any of the 64 ASCII characters from decimal 32 through 95, inclusive. The message may be as long as necessary to fill up the 12 characters in the display. Note that the period, comma, and semicolon go between characters. This command is terminated by any control character such as a <CR>. If more than 12 characters are sent, the extra characters are ignored until a control character is received. If the terminating control character is anything but an HT, LF, VT, FF, or CR, it will result in a syntax error. This command also locks the display until an error occurs, a D1 command is received, a device clear is received, or a front panel key is pressed.
	D3text	Like the D2 command above, this places the message "text" into the display. It also turns off all dedicated annunciators and then stops updating the display. This command takes about 30 mS to complete, after which the considerable overhead of updating the display is bypassed (means a faster reading rate). This enables the 3478A to respond to commands more rapidly in certain conditions. If the display is not updated for about 10 minutes, it will blank out completely. The display can be restored by pressing the "LOCAL" key, or by executing a D1, D2, or D3 command.

HARDWARE: INSTRUMENT CHARACTERISTICS

DMM CHARACTERISTICS

3478A HP-IB PROGRAMMING COMMANDS
REPRINTED WITH PERMISSION, HEWLETT PACKARD CO.

Type	Command	Explanation
Preset Commands	H0	Home Command. This command places the 3478A into the dc volts function, autorange, hold trigger state, 4½ digit display, with auto zero on. Any result ready to be output to the bus or display is erased. The external trigger input is disabled. This command is equivalent to "F1T4R−2RAZ1N4".
	H1	Measure DC Volts. This command causes one dc voltage measurement to be made, following which an enter command from the controller will access the result. This command is like the H0 command except that a trigger occurs. This command is equivalent to "F1R−2RAZ1N4T3".
	H2	Measure AC Volts. Identical to H1 except for the function.
	H3	Measure 2-Wire Ohms. Identical to H1 except for the function.
	H4	Measure 4-Wire Ohms. Identical to H1 except for the function.
	H5	Measure DC Current. Identical to H1 except for the function.
	H6	Measure AC Current. Identical to H1 except for the function.
	H7	Measure Extended Ohms. Identical to H1 except for the function.
Binary Status	B	Binary Status. If the 3478A is addressed to talk after reception of the "B" command, it will output five bytes which indicate its currently programmed state. This command also clears the error register (byte 4). The meaning of the five bytes is: Byte 1: Function, Range, and Number of Digits If Octal value of bits 7,6,5 = 1 then dc volts = 2 then ac volts = 3 then 2-wire ohms = 4 then 4-wire ohms = 5 then dc current = 6 then ac current = 7 then extended ohms If Octal value of bits 4,3,2 = 1 then 30mV dc, 300mV ac, 30Ω, 300mA ac or dc, extended ohms = 2 then 300mV dc, 3V ac, 300Ω,3A ac or dc = 3 then 3V dc, 30V ac, 3kΩ = 4 then 30V dc, 300V ac, 30kΩ = 5 then 300V dc, 300kΩ = 6 then 3MΩ = 7 then 30MΩ If Octal value of bits 1,0 = 1 then 5½ digit mode = 2 then 4½ digit mode = 3 then 3½ digit mode Byte 2: Status Bits Bit 7 = Always zero Bit 6 = 1 then External Trigger enabled Bit 5 = 1 then Cal RAM enabled Bit 4 = 1 then Front/Rear switch is in the front position Bit 3 = 1 then 3478A is set up for 50 Hz operation Bit 2 = 1 then Auto Zero is enabled Bit 1 = 1 then Autorange is enabled Bit 0 = 1 then Internal Trigger is enabled Byte 3: Serial Poll Mask (SRQ) Bit 7 = 1 then PON SRQ switch was on the last time power was turned on or a device clear message was received Bit 6 = Always zero Bit 5 = 1 then SRQ if CAL procedure failed Bit 4 = 1 then SRQ if keyboard SRQ is pressed Bit 3 = 1 then SRQ if hardware error occurs Bit 2 = 1 then SRQ if syntax error occurs Bit 1 = Not used Bit 0 = 1 then SRQ as every reading is available to the bus

HARDWARE: INSTRUMENT CHARACTERISTICS

DMM CHARACTERISTICS

3478A HP-IB PROGRAMMING COMMANDS
REPRINTED WITH PERMISSION, HEWLETT PACKARD CO.

Type	Command	Explanation
Binary Status (Cont'd)	B	Byte 4: Error Information Bit 7 = Always zero Bit 6 = Always zero Bit 5 = 1 then there has been a failure in the A/D link Bit 4 = 1 then the A/D has failed its internal self-test Bit 3 = 1 then there has been an A/D slope error Bit 2 = 1 then the ROM self-test has failed Bit 1 = 1 then the RAM self-test has failed Bit 0 = 1 then this bit is set by the self-test routine if any of the CAL RAM locations have bad checksums, or if a range with a bad checksum is selected Byte 5: DAC Value This byte will contain a value between 0 and 63 (decimal). This represents the setting of the internal Digital to Analog Converter (DAC), and is primarily for diagnostic purposes.
	K	Clear the Serial Poll Register. This command clears bits 1-5 and 7 in the Serial Poll Register. Bit 0 always represents the current status and is unaffected by this command. Bit 6 is either set (1) or cleared (0) depending upon the binary AND of bit 0 in the status register and the SRQ mask. Bit 6 is reset by a Serial Poll.
	E	Error Register. This command enables the controller to read the 3478A Error Register. See Byte 4 in the "B" command above for an explanation of this register. Unlike the "B" command, the "E" command outputs the Error Register as two octal digits, followed by a CR, LF. The "E" command also clears the Error Register.
	Mxx	Set the SRQ Mask to octal value xx. The parameter xx must be two digits exactly. Bits 0 through 5 of the binary representation of xx are used to set the appropriate mask bits.
	S	Return the Front/Rear switch position. If the 3478A is addressed to talk after receiving the "S" command, it will output either a "1" or a "0" <CR> <LF> depending on whether the front/rear switch was in the front or rear position, respectively. The position of the switch is read only when a reading is taken, and thus returns the position of the switch at time of the last reading.
CALIBRATE	C	Calibrate. See the 3478A Service Manual.

HARDWARE: INSTRUMENT CHARACTERISTICS

DMM CHARACTERISTICS

RECEIVING DATA FROM THE DMM

MEASUREMENT DATA

MEASUREMENT DATA IS READ INTO A VARIABLE USING THE ENTER COMMAND. EXAMPLE:

```
OUTPUT 723;"F1R1T2" ! DC,30VFS,EXT TRIG
ENTER 723;VOLTS ! READ INTO VARIABLE VOLTS
PRINTER IS 705 ! INITIALIZE PRINTER
PRINT " THE READING IS " ! MESSAGE PRINTED
PRINT VOLTS ! PRINTS READING TAKEN
```

PLACE THE TRIGGER AT THE END OF THE COMMAND STRING AS THE COMMANDS ARE EXECUTED AS THEY ARRIVE AT THE DMM. EXAMPLE: THE PROGRAM BELOW WILL NOT FUNCTION AS COMMENTED. ASSUME THE PRIOR SET UP WAS F1-DC,R0-3VFS,T3-SINGLE TRIGGER. VOLTAGE IS 10-15V.

```
10 OUTPUT 723;"T2F1R1" ! EXT TRIG,DC,30VFS
20 FOR N=1 to 10 ! 10X LOOP
30 ENTER 723;V(N) ! FILL 10 ARRAY LOCATIONS
40 NEXT N ! LOOP BACK OR NOT
50 PRINTER IS 705 ! INITIALIZE PRINTER
60 OUTPUT 705;(*) ! SEND WHOLE ARRAY TO PRINTER
70 END
```

WHAT WILL BE LOADED INTO THE ARRAY AND PRINTED?

WRITE A SHORT PROGRAM WHICH WILL ALTERNATE IN ENTERING FIVE READINGS EACH FROM TWO DMMs INTO ONE ARRAY. DMM ADDRESSES:723, 725. KEEP COMMAND LINES SHORT! WRITE PROGRAM HERE.

SOFTWARE: BASIC ENHANCEMENTS

FUNDAMENTAL STATEMENTS

REMARK (COMMENT) STATEMENT - IN MANY CASES SOFTWARE HAS TO BE EVALUATED FROM SEVERAL WEEKS TO SEVERAL YEARS LATER. YOU OR SOMEONE ELSE MAY HAVE THE TASK OF HAVING TO FOLLOW THE LOGIC OF THE PROGRAM. THESE COMMENTS SHOULD BE WRITTEN AS TO MAKE THE LOGIC OF THE PROGRAM EASY TO FOLLOW.

FORMAT

```
REM (any set of characters)
```

THE EXCLAIMATION POINT ! MAY BE USED IN PLACE OF THE REM STATEMENT.

EXAMPLE:

```
10 REM ***** COMPUTER DOES NOT READ COMMENT *****

20 PRINT A$ ! A$ IS THE VOLTAGE READING AT TP3

                  ONE SPACE
```

DISPLAY STATEMENT - THIS STATEMENT WILL OUTPUT VARIABLE VALUES OR STRING CHARACTERS TO THE CRT OR TO THE DEVICE SPECIFIED IN THE CRT IS STATEMENT.

FORMAT

```
DISP {item {,item...}}
     {item {;item...}}
```

EXAMPLE:

IF A=2.45 AND B$ = " THE VALUE IS " THEN AFTER THE EXECUTION OF DISP B$,A THE FOLLING IS RETURNED.

```
THE VALUE IS          2.45    i.e. FREE FIELD
```

DISP B$;A WILL RETURN

```
THE VALUE IS 2.45      i.e. COMPACT FIELD
```

NOTES:

SOFTWARE: BASIC ENHANCEMENTS

FUNDAMENTAL STATEMENTS

PRINT STATEMENT - VARIABLE VALUES AND STRING CHARACTERS ARE OUTPUT TO THE PRINTER SPECIFIED IN "THE PRINTER IS" STATEMENT. IF NO PRINTER IS SPECIFIED THEN THE DATA IS OUTPUT TO THE CRT.

FORMAT

```
PRINT {item {,item}}
      {item {;item}}
```

EXAMPLE:

```
PRINT "AMPLITUDE VALUE";V     WILL RETURN

AMPLITUE VALUE 34.7      i.e. V=34.7
```

INPUT STATEMENT - THIS STATEMENT ALLOWS ENTRY OF DATA INTO A RUNNING PROGRAM. SEVERAL DATA ITEMS MAY BE ENTERED WITH ONE INPUT STATEMENT.

FORMAT

```
INPUT variable name {,variable name...}
```

EXAMPLE:

AFTER THE PROGRAM EXECUTES INPUT A,B,C THE FOLLOWING IS DISPLAYED ON THE CRT.

```
?
YOU ENTER       data,data,data
```

BEEP STATEMENT - THIS STATEMENT CAUSES AN AUDIBLE OUTPUT FROM THE COMPUTER.

FORMAT

```
BEEP {tone,duration}
```

EXAMPLE:

EXECUTION OF BEEP RETURNS A 2KHz. 100 mS BEEP

WAIT STATEMENT - THIS STATEMENT WILL INTRODUCE A DELAY BETWEEN PROGRAM STATEMENTS. THIS CAN BE USED TO INTRODUCE DELAYS IN MEASUREMENT SYSTEMS TO ALLOW FOR SETTLING TIMES.

FORMAT

```
WAIT milliseconds
```

EXAMPLES:

```
WAIT 500   WILL INTRODUCE A 1/2 SECOND DELAY
```

NOTES:

FUNDAMENTAL STATEMENTS

READ AND DATA STATEMENTS - THE NUMBERS OR CHARACTERS WITHIN THE DATA STATEMENT ARE ASSIGNED TO THE VARIABLES IN THE READ STATEMENT. THESE VARIABLES MAY BE NUMERIC, STRING, OR ARRAY TYPES. IF MORE READ VARIABLES EXIST THAN DATA, THEN AN ERROR WILL BE GENERATED.

FORMAT

```
READ variable name{,variable name...}

DATA constant{,constant...}
```

EXAMPLE:

```
10 READ D1,D2,D3
20 DISP "SETTING 1 IS ";D1
30 DISP "SETTING 2 IS ";D2
40 DISP "SETTING 3 IS ";D3
50 DATA 2.5,6.4,8.2
```

AFTER EXECUTION OF THE PROGRAM, THE FOLLOWING IS DISPLAYED:

```
SETTING 1 IS 2.5
SETTING 2 IS 6.4
SETTING 3 IS 8.2
```

RESTORE STATEMENT - THIS ALLOWS DATA ITEMS TO BE REUSED.

FORMAT

```
RESTORE {line number}
        {statement label}
```

EXAMPLE:

```
10 OUTPUT 707; "FU1 FR1KZ AP1VO"
20 FOR N = 1 TO 4 ! DECADE INCREMENT LOOP
30 FOR M = 1 TO 3 ! INTEGER INCREMENT LOOP
40 READ X ! BRING IN INTEGER
50 OUTPUT 707;"FR",(X*E(N)),"KZ" ! OUTPUT NEW FREQUENCY
60 NEXT M ! LOOP BACK TO NEXT INTEGER
70 RESTORE 90
80 NEXT N ! LOOP BACK TO NEXT POWER
90 DATA 2,4,8 ! INTEGER VALUES
100 END
```

NOTES:

FUNDAMENTAL STATEMENTS

END AND STOP STATEMENT - THESE STATEMENTS TERMINATE PROGRAM EXECUTION AND MAY BE USED INTERCHANGEABLY. THEY ARE OFTEN THE LAST LINE IN A PROGRAM, ALTHOUGH THEY MAY APPEAR EARLIER IN THE PROGRAM WHERE THEY MAY BE BYPASSED BY BRANCHES (GOTO, IF THEN) OR SUBROUTINES (GOSUB).

FORMAT

```
END
STOP
```

EXAMPLES:

```
10 MAIN BODY OF PROGRAM
.
.
130 IF.......THEN 150
140 END
150 PROGRAM CONTINUES
.
.
240 GOSUB 540
250 PROGRAM CONTINUES
.
310 END
540 SUBROUTINE STARTS HERE
.
.
630 RETURN
```

NOTES:

BRANCHES AND LOOPS

GOTO STATEMENT - THIS STATEMENT IS AN UNCONDITIONAL JUMP, THAT IS THE PROGRAM WILL BRANCH TO THE LOCATION SPECIFIED IN THE GOTO STATEMENT UNDER ALL CONDITIONS. THE GOTO IS NOT AS VERSATILE AS THE GOSUB AND SHOULD NOT BE USED IN ITS PLACE.

FORMAT

```
GOTO line number
     statement label
```

EXAMPLE:

```
10 ENTER 723; A ! PLACE READING INTO VARIABLE A
20 WAIT 3600 ! DELAY ONE MINUTE
30 PRINT A ! LOG THE VALUE OF "A" ON A PRINTER
40 GOTO 10 ! LOOP BACK FOREVER OR UNTIL RESET OR PAUSE
```

ON GOTO STATEMENT - THIS IS ALSO AN UNCONDITIONAL BRANCH WHICH WILL BRANCH TO ONE OF SEVERAL SPECIFIED LOCATIONS BASED ON THE OUTCOME OF A NUMERIC EXPRESSION. THE ROUNDED VALUE OF THE NUMERIC RESULT WILL DETERMINE WHICH LINE NUMBER WILL BE BRANCHED TO i.e. 1st,2nd,ETC.

FORMAT.

```
ON numeric expression GOTO line number
                           statement label
```

EXAMPLE:

```
10 ENTER 723; V
20 ON V/10 GOTO 30,40,50
30 DISP "SELECT RANGE 1"
40 DISP "SELECT RANGE 2"
50 DISP "SELECT RANGE 3"
.
.
230 END
```

IF THEN STATEMENT - THIS IS A CONDITIONAL BRANCH STATEMENT WHICH WILL BRANCH IF A CONDITION IS MET OR CONTINUE SEQUENTIAL PROCESSING.

FORMAT

```
IF logical expression THEN line number
   numerical expression    statement label
                           executable expression
```

EXAMPLE:

```
10 ENTER 723; V
20 IF V<>9.99999+9 THEN 50
30 DISP "OVER RANGE CONDITION DO NOT PROCEED"
40 END
50 ! PROGRAM CONTINUES HERE IF NOT OVER RANGE.
```

SOFTWARE: BASIC ENHANCEMENTS

BRANCHES AND LOOPS

IF THEN ELSE STATEMENT - IN THE IF THEN STATEMENT IF IT WAS FALSE THEN PROGRAM EXECUTION WOULD PROCEED SEQUENTIALLY. THE ELSE OPTION ALLOWS ALTERNATE OPERATIONS ON THE FALSE CONDITION.

FORMAT

```
IF numeric expression THEN line number ELSE line number
   logical expression      statement label  statement label
                           executable stmt. executable stmt.
```

EXAMPLE:

```
10 DISP "ENTER UUT MODEL NUMBER " ! VALUE IS 5 OR 15
20 INPUT A
30 IF A=5 THEN MOD5 ! CHECKS FOR MODEL 5
40 IF A=15 THEN GOSUB MOD15 ELSE 70 ! CHECKS FOR MODEL 15
50 DISP "TEST OVER, PRESS RUN TO REPEAT" ! REPEAT TEST
60 END
70 DISP "RE-ENTER MODEL NUMBER" ! WRONG MODEL ENTERED
80 GOTO 20
90 MOD5: ! MODEL 5 TEST IS HERE
.
.
320 GOTO 50 ! TO TEST OVER MESSAGE
330 MOD15: MODEL 15 TEST SUBROUTINE IS HERE
.
.
670 RETURN ! TO TEST OVER MESSAGE
```

FOR NEXT LOOPS - PROGRAM LOOPS ARE CREATED WITH THE FOR-NEXT STATEMENTS. VARIABLES WITHIN THE LOOP MAY BE ALTERED BY THE LOOP COUNTER.

FORMAT

```
FOR loop counter = numeric expression TO num. exp.{STEP num.ex}

NEXT loop counter
```

EXAMPLE:

```
10 OUTPUT 707; "PR AP1VO" ! SET UP SINE, 1KHz, AMPLITUDE 1Vpp
20 FOR N = 1000 TO 50000 STEP 2000 ! SET LOOP COUNTER
30 WAIT 500 ! CIRCUIT SETTLING TIME
40 ENTER 723; V ! TAKE READING
50 PRINT 705; V ! PRINT VALUE JUST READ
60 OUTPUT 707; "FR",N,"HZ" ! INCREASE FREQUENCY BY 2KHz
70 NEXT N
80 ! PROGRAM CONTINUES
```

NOTES:

ARRAYS

PRELUDE - WHEN MANY DATA ITEMS OF THE SAME TYPE ARE NEEDED IT IS IMPRACTICAL TO ASSIGN A VARIABLE NAME TO EACH. TO ACCOMODATE LARGE AMOUNTS OF DATA AN ARRAY IS SET UP. THUS MANY DATA ITEMS OF THE SAME TYPE CAN BE GROUPED UNDER THE SAME NAME. DATA MAY BE NUMERIC OR STRING.

ONE DIMENSIONAL ARRAYS - A LIST OF DATA ITEMS, A SINGLE ROW.

TWO DIMENSIONAL ARRAYS - A MATRIX OF DATA ITEMS ORGANIZED IN A ROW - COLUMN ARRANGEMENT.

BOUNDS OF AN ARRAY - DETERMINES NUMBER OF DATA ITEMS IN THE ARRAY (ELEMENTS)

LOWER BOUND - ELEMENT WITH LOWEST SUBSCRIPT (DEFAULT IS 0)

UPPER BOUND - ELEMENT WITH HIGHEST SUBSCRIPT (IF SUBSCRIPT > 10, THE ARRAY MUST BE DIMENSIONED.)

OPTION BASE - USE TO SET THE LOWER BOUND

FORMAT

```
OPTION BASE 1   note: only 0 or 1
```

NUMERIC ARRAY EXAMPLES:

DATA IN ARRAY R2 IS 5,7,23,16

R2(0) IS 5

R2(3) IS 16

```
DATA IN ARRAY VOLTS IS 8,4,2
                       5,7,9
                       6,3,1
```

VOLTS(1,1) IS 7

VOLTS(0,0) IS 8

VOLTS(2,1) IS 3

VOLTS(1,2) IS ?

ARRAY SUBSCRIPT - MAY BE EITHER A NUMERIC VALUE OR A NUMERIC VARIABLE

GAIN(5,3)

GAIN(N,M)

NOTES:

SOFTWARE: BASIC ENHANCEMENTS

ARRAYS

DIMENSIONING A NUMERIC ARRAY - AN ARRAY MUST BE DIMENSIONED, FOR THE COMPUTER TO ALLOCATE SPACE FOR IT, IF ANY OF ITS SUBSCRIPT VALUES ARE > 10.

FORMAT

```
DIM numeric variable name(upper bound{,upper bound})
```

EXAMPLE:

DIM R2(5) SETS UP A ONE DIMENSIONAL ARRAY WITH AN UPPER BOUND OF 5.

DIM PKVAL(100,20) SETS UP A TWO DIMENSIONAL ARRAY WITH 101 ROWS AND 21 COLUMNS.

HP SERIES 80 COMPUTERS ALLOW FURTHER DEFINITION OF THE PRECISION OF THE NUMERIC VALUE.

REAL - (SAME AS DIM) NUMBERS ARE STORED IN THE FULL PRECISION OF THE COMPUTER i.e. 12 DIGIT MANTISSA AND 3 DIGIT EXPONENT.

SHORT - NUMBERS ARE DIMENSIONED IN SHORT PRECISION i.e. 5 DIGIT MANTISSA AND 2 DIGIT EXPONENT.

INTEGER - DIMENSIONS NUMBERS IN INTEGER PRECISION i.e. STORED AS 5 DIGIT INTEGER VALUES, NO DIGITS BEYOND THE DECIMAL POINT.

COM - ALLOWS DATA TO BE COMMON BETWEEN TWO SEPERATE PROGRAMS WHICH ARE LINKED TOGETHER WITH THE CHAIN COMMAND. THIS IS THE ONLY WAY DATA CAN BE PASSED BETWEEN PROGRAMS i.e. BE COMMON TO BOTH PROGRAMS. ONLY USE CHAINED PROGRAMS WHEN MEMORY SPACE WILL NOT ACCOMODATE ALL THE SUBROUTINES NEEDED.

FORMAT

```
COM {precision} item
```

EXAMPLE:

```
COM TEST(4,200)

COM INTEGER TEST(4,200)
```

NOTES:

SOFTWARE: BASIC ENHANCEMENTS

CHAINING PROGRAMS

CHAIN STATEMENT - THIS STATEMENT ALLOWS ONE PROGRAM TO LOAD AND RUN ANOTHER PROGRAM. ALL THE DATA IN THE CURRENT BASIC PROGRAM WILL BE SCRATCHED WHEN THE NEW PROGRAM IS CHAINED (LOADED). THE COM STATEMENT IS THE ONLY WAY TO PRESERVE DATA BETWEEN PROGRAMS. VARIALE NAMES NEED NOT BE IDENTICAL IN THE PROGRAMS CHAINED. THE COM VARIABLES WILL BE USED IN THE ORDER THEY ARE PRESENT IN THE PROGRAM.

EXAMPLE:

```
10 REM ** MAIN PROGRAM (MAINPRG) **
20 COM M$(12) ! DIMENSIONS STRING TO 12 CHARACTERS
30 DISP "ENTER UP TO 12 DIGIT DMM MESSAGE" ! MESSAGE ON CRT
40 INPUT M$ ! INPUT MESSAGE TO BE DISPLAYED
50 CHAIN "DISPPRG" ! LOADS AND RUNS DISPLAY PROGRAM
60 END ! END OF MAIN PROGRAM
```

```
10 REM ** DISPLAY PROGRAM (DISSPRG) **
20 COM DMMMSG$(12) ! COMMON STRING VARIABLE NAME CHANGED
30 OUTPUT 723: "D2"&DMMMSG$ ! STRING IS CONCATENATED TO D2
40 END
```

THE DATA (UP TO 12 CHARACTERS) ENTERED FOLLOWING THE INPUT STATEMENT IS DISPLAYED ON THE ALPHA- NUMERIC DISPLAY OF THE DMM.

NOTES:

STRING ARRAYS

DIMENSIONING A STRING ARRAY - IF A STRING ARRAY IS GOING TO REQUIRE AN UPPER BOUND >10 AND CHARACTERS PER ARRAY ELEMENT >18 IT MUST BE DIMENSIONED.

FORMAT

DIM string variable name (upper bound{,upper bound}){length per element}

EXAMPLES:

DIM TEST$
SETS UP A ONE DIMENSIONAL ARRAY OF 10 ELEMENTS AND 18 CHARACTERS PER ELEMENT.

DIM TEST$(20){22}
SETS UP A ONE DIMENSIONAL ARRAY OF 20 ELEMENTS AND 22 CHARACTERS PER ELEMENT.

DIM TEST$(20,15){12}
SETS UP A TWO DIMENSIONAL ARRAY OF 20 ROWS X 15 COLUMS WITH 12 CHARACTERS PER ELEMENT.

STRING OPERATIONS

SUBSTRING - A PORTION OF A STRING

FORMAT

string variable name{beginning character position{,ending character position}}

EXAMPLE:

IF STRING V$ CONTAINS THE FOLLOWING DATA: +3.47651E-3CrLf THEN THE FOLLOWING STATEMENT WILL CAUSE +3.47651E-3 TO BE DISPLAYED.

```
DISP V${1,11}
```

LENGTH STATEMENT - WILL RETURN THE NUMBER OF CHARACTERS IN A STRING.

FORMAT

LEN(string expression)

EXAMPLE:

IF R2$ CONTAINS 475, THEN LEN(R2$) WILL RETURN 3. THUS THE NUMBER OF CHARACTERS IN THE STRING MAY BE DETERMINED.

NOTES:

STRING ARRAYS

STRING OPERATIONS

VALUE FUNCTION - THIS FUNCTION IS USED TO CONVERT STRING NUMBERS TO THEIR NUMERIC VALUE, SO THEY MAY BE USED IN NUMERIC OPERATIONS.

FORMAT

```
VAL(string expression)
```

EXAMPLE:

IF V$ CONTAINS "READING 1 IS 25", THEN VAL(V${13}) WILL RETURN THE VALUE STARTING FROM THE 13th CHARACTER IN THE STRING i.e. 25.

THE FIRST CHARACTER MUST BE A DIGIT, PLUS OR MINUS SIGN, A DECIMAL POINT, OR A SPACE.

POSITION FUNCTION - THIS FUNCTION RETURNS THE POSITION OF ONE STRING WITHIN ANOTHER STRING.

FORMAT

```
POS(string 1,string 2)    i.e. POSITION OF STRING 2 IN 1
```

EXAMPLE:

IF A$= +3.45654E1 AND B$=. THEN POS(A$,B$) WILL RETURN 3. THIS CAN BE USED TO DETERMINE HOW MANY INTEGERS ARE IN A$.

NUMERIC FUNCTION - THIS FUNCTION RETURNS THE ASCII VALUE FOR THE FIRST CHARACTER OF THE SPECIFIED STRING.

FORMAT

```
NUM(string expression)
```

EXAMPLE: THE FOLLOWING PROGRAM DETERMINES IF THE 12th CHARACTER OF A STRING IS A CARRIAGE RETURN - Cr.

```
10 ENTER 723; A ! ENTERS VALUE INTO NUMERIC VARIABLE
20 A$=VAL$(A) ! CONVERTS NUMERIC VARIABLE TO STRING VARIABLE
30 B$=A${12,12} ! SUBSTRING B$=12th CHARACTER OF STRING A$
40 C=NUM(B$) ! ASCII VALUE IS DETERMINED FOR 12th CHARACTER
50 IF C=13 THEN 80 ! IF Cr i.e. 13, THEN CARRY ON
60 DISP "NO Cr TRANSMITTED OPERATION ABORTED"
70 END
80 ! PROGRAM RESUMES HERE
```

SOFTWARE: BASIC ENHANCEMENTS

FORMATTING DATA FOR DISPLAY OR PRINT

FORMAT STRING - THE FUNCTION OF THIS STRING IS TO FORMAT, OR DESCRIBE DESCRIBE HOW THE DATA WILL BE DISPLAYED. THE FORMAT STRING CONSISTS OF THE FOLLOWING ITEMS:

- FIELD SPECIFIER - DEFINES THE FORMAT OF EACH FIELD
- IMAGE SPECIFIER - DEFINES THE FORMAT WITHIN EACH FIELD

EXAMPLE:

```
                          FORMAT STRING

                FIELD   IMAGE

100 PRINT USING "6A,2X,A,2X,2/,8A,SD.D,X,SD.D";"LINE 1",A,B,2.3,3.4
```

THE FOLLOWING IS PRINTED FOLLOWING THE EXECUTION OF LINE 100

```
LINE 1  A    B

        2.3  3.4
```

REPRINTED WITH PERMISSION COURTESY OF HEWLETT PACKARD CO.

Type of Output	Character	Purpose	Replication
Blank space	X	Specifies a blank space between items.	Yes
Radix symbol	.	Specifies a decimal point in that position.	No
	R	Specifies a comma radix indicator in that position.	No
Digit	D	Digit position to left or right of radix symbol; leading blanks.	Yes
	Z	Digit position to left of radix symbol; leading zeros.	Yes
	*	Digit position to left of radix symbol; leading asterisks (*).	Yes
Digit separator	C	Specifies a comma as a separator in the specified position.	No
	P	Specifies a period as a separator in the specified position.	No
Exponent	E	Numeric field is output in exponential format; exponent consists of three digits plus sign.	No
Sign	S	Specifies sign, "+" or "−".	No
	M	Specifies sign, blank or "−".	No
String	" "	Specifies literal text.	No
	A	Specifies character position; text is left-justified.	Yes
Numeric or string	K	Specifies compact format with no leading or trailing blanks.	No
Miscellaneous	( )	Used to indicate replication of field specifier.	Yes
	/	Specifies a carriage return/line feed; can also be used as a delimiter.	Yes

NOTES:

FORMATTING DATA FOR DISPLAY OR PRINT

IMAGE STATEMENT - USED TO SPECIFY A FORMAT STRING WHERE THE USE OF A FORMAT STRING WOULD BE UNDESIREABLE TO SPECIFY IT IN THE PRINT STATEMENT. THIS WOULD OCCUR WHERE THE FORMAT STRING MUST BE USED MANY TIMES.

FORMAT

```
IMAGEformat string
```

EXAMPLE:

```
10 IMAGE"VOLTS",2X,SD.D,//
20 PRINT USING 10;2.3,3.6,-8.9
```

THE FOLLOWING IS PRINTED FOLLOWING EXECUTION OF LINE 20.

```
VOLTS +2.3

VOLTS +3.6

VOLTS -8.9
```

PRINT USING AND DISPLAY USING STATEMENTS MAY OCCUR THROUGHOUT THE PROGRAM AND REFERENCE THE IMAGE STATEMENT.

TAB FUNCTION - THE TAB FUNCTION CAN BE APPENDED TO PRINT AND DISP STATEMENTS TO SPECIFY THE COLUMN POSITION AT WHICH DATA WILL BE PRINTED OR DISPLAYED.

FORMAT

```
TAB(character position)
```

EXAMPLE:

```
.
.
230 FOR N=0 TO 7
240 PRINT TAB (10*N):VOLT(N)
250 NEXT N
260 ! PROGRAM CONTINUES HERE
```

NOTES:

SUBROUTINES

GOSUB STATEMENT - THIS BRANCHES THE PROGRAM TO THE SUBROUTINE SPECIFIED. USED WHERE THE FOR NEXT LOOP IS INAPPROPRIATE i.e. THOSE ROUTINES THAT ARE USED FREQUENTLY, BUT NOT SEQUENTIALLY.

FORMAT

```
GOSUB statement label
      line number

RETURN    EACH GOSUB MUST HAVE A RETURN
```

NESTED SUBROUTINES - SUBROUTINES WITHIN SUBROUTINES. UP TO 255 LEVELS ARE ALLOWED, EACH MUST HAVE A CORROSPONDING RETURN.

ON GOSUB - THIS TYPE OF OPERATION IS KNOWN AS A COMPUTED SUBROUTINE. IT ALLOWS BRANCHING TO ONE OF SEVERAL SUBROUTINES BASED ON THE INTEGER RESULT OF A NUMERIC EXPRESSION. (LIKE ON GOTO)

FORMAT

```
ON numeric expression GOSUB line number{,line number...}
                            statement label{,statement label}
```

USER DEFINED FUNCTIONS

A SINGLE OR MULTILINE FUNCTION (MACRO) MAY BE DEFINED BY THE USER AND CALLED UPON BY THE FUNCTIONS NAME.

FORMAT

```
DEF FNnumeric function name{(parameter{,parameter...})}=num expres

DEF FNstring function name{(parameter{,parameter...})}=str express
```

16 NUMERIC OR 7 STRING PARAMETERS MAY BE SPECIFIED

EXAMPLES:

```
10 DEF FNZ(A,B)=(A*A+B*B) .5

10 DEF FNTEST$(A)= "RESULTS ARE "&VDC$ (A)
```

WHEN REFERENCED IT INCLUDES NAME AND ARGUMENTS

FORMAT

```
FNnumeric function name{(argument{,argument...})}
  string function name
```

EXAMPLES:

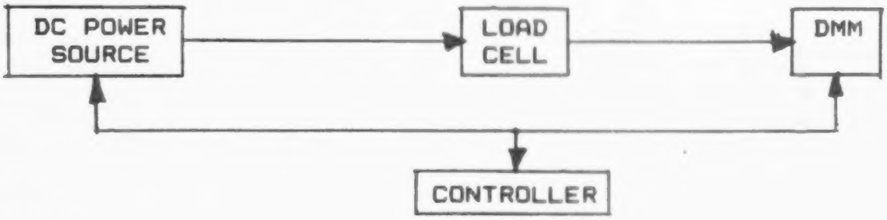
```
50 Iin=Vin/FNZ(R,Xc) ! COMPUTES Iin FROM Vin AND Z

50 B=105 @ DISP FNTEST$(B) ! RESULTS ARE 105 WILL BE DISPLAYED
```

NOTES:

COMPUTER CONTROLLED TEST SYSTEMS

LAB PROJECT 2: ELECTRONIC SCALE



1. THE PURPOSE OF THIS PURPOSE OF THIS LAB IS TO DEVELOP THE FOLLOWING THREE APPLICATION PROGRAMS WHICH WILL UTILIZE A LOAD CELL AS THE DEVICE BEING MONITORED BY THE BUS.

A. PARTS COUNTING SCALE, WHICH WILL DETERMINE THE WEIGHT OF ONE ITEM BY MEASURING ONE OR SEVERAL. IT WILL THEN USE THIS TO COUNT THOSE ITEMS AS THEY ARE PLACED ON THE SCALE.

B. METRIC WEIGHING SCALE, WHICH WILL WEIGH ITEMS BETWEEN 1 GRAM AND 1 Kg ONCE CALIBRATED.

C. OPTIONAL: MONETARY COUNTING SCALE, DETERMINE IF THE LOAD CELL IS SENSITIVE ENOUGH TO WEIGH COINS. THE PROGRAM SHOULD ADD UP THE TOTAL COINS ADDED IN DOLLARS AND CENTS.

2. IN ORDER TO REDUCE THE EFFECTS OF NOISE ON VOLTAGE READINGS SEVERAL READINGS SHOULD BE AVERAGED. THE EFFECT OF NOISE IS REDUCED BY $1/\sqrt{N}$ FOR N SAMPLES.

3. THE PROGRAM SHOULD INCLUDE THE FOLLOWING FUNCTIONS:

- POWER UP SELF TEST
- DETERMINES LOCATION OF DMM ON THE BUS

4. PREPARE A REPORT WHICH CONTAINS THE FOLLOWING INFORMATION:

- DETAILS OF YOUR PROGRAM OPERATION
- PROGRAM FLOW CHART
- ACCURACY OF YOUR MEASUREMENT METHOD, i.e. CAPABILITIES

5. THIS PROJECT MUST BE COMPLETED IN ORDER TO RECEIVE A B GRADE IN THIS CLASS.

6. SECURE THIS PROGRAM ON YOUR STORAGE MEDIUM TO PREVENT UNAUTHORIZED COPYING. USE LABPR2 FOR FILE SPECIFIER, USE THE LAST TWO DIGITS OF YOUR STUDENT NUMBER FOR THE SECURITY CODE, USE SECURITY TYPE 3.

SOFTWARE: BASIC STRUCTURED PROGRAMMING

PRELUDE - STRUCTURED PROGRAMMING IS A METHOD OF PROGRAM DEVELOPMENT WHICH FOLLOWS A SET OF LOGICALLY PREDEFINED OPERATIONS AND METHODS. THE OVERALL TASK IS BROKEN DOWN INTO A SET OF SUB TASKS WHICH ARE DESIGNED AS AUTONOMOUS OPERATIONS TO THEMSELVES. THIS MEANS THAT EACH SUB TASK SHOULD BE ABLE TO PERFORM ITS INTENDED FUNCTION WHEN PASSED ITS REQUIRED DATA, WHETHER IT IS ASSOCIATED WITH THE TASK IT WAS ORIGIONALLY DESIGNED FOR OR NOT.

THE FRAMEWORK AND GENERAL CONCEPTS WHICH FOLLOW HAVE BEEN EXTRACTED FROM THE HEWLETT PACKARD DOCUMENT "HP-IB INSTRUMENT CONTROL USING HP SERIES 200 BASIC".

BENEFITS OF STRUCTURED PROGRAMMING

* EASY TO READ AND UNDERSTAND

* EASY TO MODIFY AND UPDATE

* EASY TO UNDERSTAND BY OTHERS

* SELF-DOCUMENTING

* STRUCTURE FORMAT IS UNIVERSAL

* FOLLOWS LOGICAL THOUGHT PROCESS

* PROGRAM DEVELOPMENT MAY BE EASILY DIVIDED BETWEEN PROGRAMMERS

* EASILY TESTED (DEBUGGED)

NOTES:

SOFTWARE: BASIC STRUCTURED PROGRAMMING

DESIGN STEPS FOR A STRUCTURED PROGRAM

1. CREATE A TEST REQUIREMENT SPECIFICATION - THIS IS A DOCUMENT WHICH REFERS TO THE ITEM BEING TESTED, NOT THE TEST ITSELF.

   A. THE PROBLEM TO BE SOLVED MUST BE WRITTEN DOWN . THE EASIEST METHOD IS TO USE THE NARRATIVE FORM.

   B. LIST THE CHARACTERISTICS OF THE DEVICE BEING TESTED i.e. THE PARAMETERS WHICH WILL BE EVALUATED IN THE TEST.

   C. FROM THE PARAMETERS OF THE UNIT UNDER TEST (UUT) TO BE EVALUATED, A LIST OF MEASUREMENT TYPES SHOULD THEN BE GENERATED i.e. VOLTAGE, PHASE , GAIN, INDUCTANCE, ETC.

   D. A LIST OF THE DATA TYPES TO BE PRODUCED SHOULD THEN BE CREATED. THE LIST SHOULD INCLUDE REQUIRED FORMAT (ASCII, BCD, INTEGER, FLOATING POINT, ETC.), REQUIRED ACCURACY (SIGNIFICANT DIGITS, TO WITHIN A SPECIFIED PERCENTAGE OF A PREDEFINED VALUE).

   E. A SET OF REQUIREMENTS WHICH DESCRIBES HOW THE UUT WILL BE EVALUATED EITHER GOOD OR BAD IS INCLUDED AS THE FINAL PORTION OF THE DOCUMENT.

2. TEST SPECIFICATION DOCUMENT - A DOCUMENT WHICH DESCRIBES THE TEST UTILIZING THE FOLLOWING COMPONENTS.

   A. METHODOLOGY USED TO EVALUATE UUT. THIS COULD BE DESCRIBED FIRST WITH A GENERAL NARRATIVE DESCRIPTION, IF THE TASK IS LENGTHLY.

   B. CHOOSE ONE OF THE FOLLOWING METHODS TO SHOW PROGRAM FLOW.

      1. FLOW CHART

      2. PSEUDOCODE

      3. WARNIER-ORR DIAGRAM

      4. NASSI-SHNEIDERMAN DIAGRAM

   C. BREAK THE PROBLEM TO BE SOLVED DOWN INTO INDIVIDUAL TASKS. EACH OF THESE TASKS SHOULD BE ABLE TO FUNCTION INDEPENDENTLY OF THE OTHER TASKS WITHIN THE PROGRAM. IT SHOULD BE THOUGHT OF AS A "BLACK BOX" WHICH HAS A PRESCRIBED FUNCTION AND PERFORMS THAT FUNCTION ON DATA WHICH IS PASSED TO IT AND RETURNS ITS RESULTS.

NOTES:

DESIGN STEPS FOR A STRUCTURED PROGRAM

D. DEVELOP A LIST OF CONSTRAINTS WHICH AFFECT HOW THE PROBLEM IS SOLVED AND CAN AFFECT THE RESULTS i.e. MEMORY CAPACITY, BANDWIDTH, INSTRUMENTS AVAILABLE, INSTRUMENT & CONTROLLER SPEED, ETC.

E. THE FOLLOWING ITEMS AND THE METHOD OF THEIR IMPLEMENTATION MUST BE INCLUDED.

1. WHAT INTERFACE(S) WILL BE USED.
2. HOW DATA WILL BE OBTAINED i.e. MEASURED OR CALCULATED
3. HOW WILL UUT BE STIMULATED i.e. SOURCE AUTOMATIC OR MANUAL
4. TYPE OF FIXTURING AND CABLEING NEEDED
5. SIGNAL SWITCHING REQUIRED

F. TO ENHANCE THE ACCURACY OF MEASUREMENTS, MULTIPLE REFERENCE MEASUREMENTS NEED TO BE TAKEN DURING THE TEST.

3. DEFINE VARIABLE NAMES, TYPES, AND SIZES

4. TEST REPORT DOCUMENT - THIS WILL CONTAIN DATA AND ANALYSIS INFORMATION ON SPECIFIC UUTs AND GROUPS OF UUTs.

A. FAILURE ANALYSIS REPORT - THIS IS A DOCUMENT THAT WILL USUALLY ACCOMPANY THE UUT AFTER THE TEST HAS BEEN COMPLETED. THIS DOCUMENT WILL CONTAIN THE FOLLOWING TYPES OF INFORMATION.

1. PART #
2. SERIAL #
3. TEST OPERATOR OR NUMBER
4. FAILURE DATA

B. BATCH (LOT, GROUP, RUN) REPORT - A REPORT ON THE OVERALL RESULTS OF THE GROUP OF ITEMS TESTED. THIS COULD INCLUDE THE FOLLOWING TYPES OF INFORMATION.

1. PART #
2. LIST OF SERIAL NUMBERS IN GROUP
3. FAILURE ANALYSIS
    A. % UUTs FAILED
    B. CORRELATION OF FAILURES

NOTES:

SOFTWARE: BASIC STRUCTURED PROGRAMMING

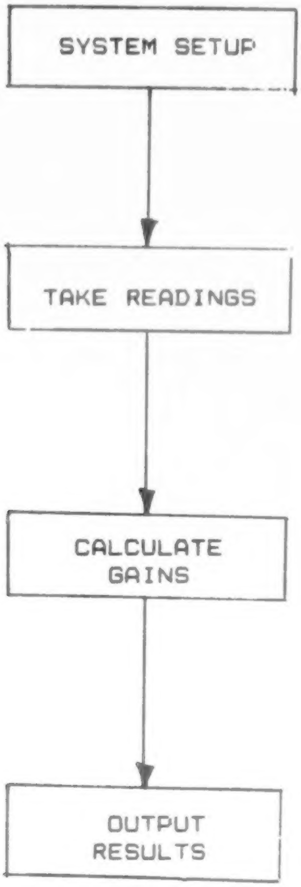
TOP-DOWN DESIGN

THIS TECHNIQUE LENDS ITSELF TO PROBLEMS WHICH ARE SEQUENTIAL IN NATURE AND HAVE A MINIMAL NUMBER OF BRANCHES.

MAY BE USED TO AS A FIRST STEP IN DESIGNING A PROGRAM.

EXAMPLE:

AMPLIFIER BANDWIDTH TEST


SYSTEM SETUP
TAKE READINGS
CALCULATE GAINS
OUTPUT RESULTS

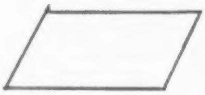
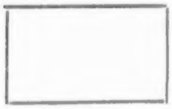
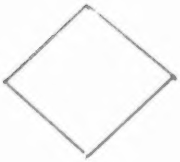
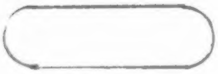
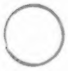
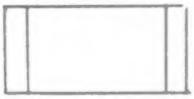
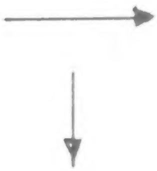
NOTES:

SOFTWARE: BASIC STRUCTURED DESIGN

FLOWCHARTING

THIS IS A TECHNIQUE WHICH EMPLOYS SYMBOLS WHICH HAVE A PREDEFINED FUNCTION. THESE SYMBOLS CONTAIN PROGRAM OPERATION DESCRIPTORS AND ARE LINKED TOGETHER WITH DIRECTIONAL FLOW LINES.

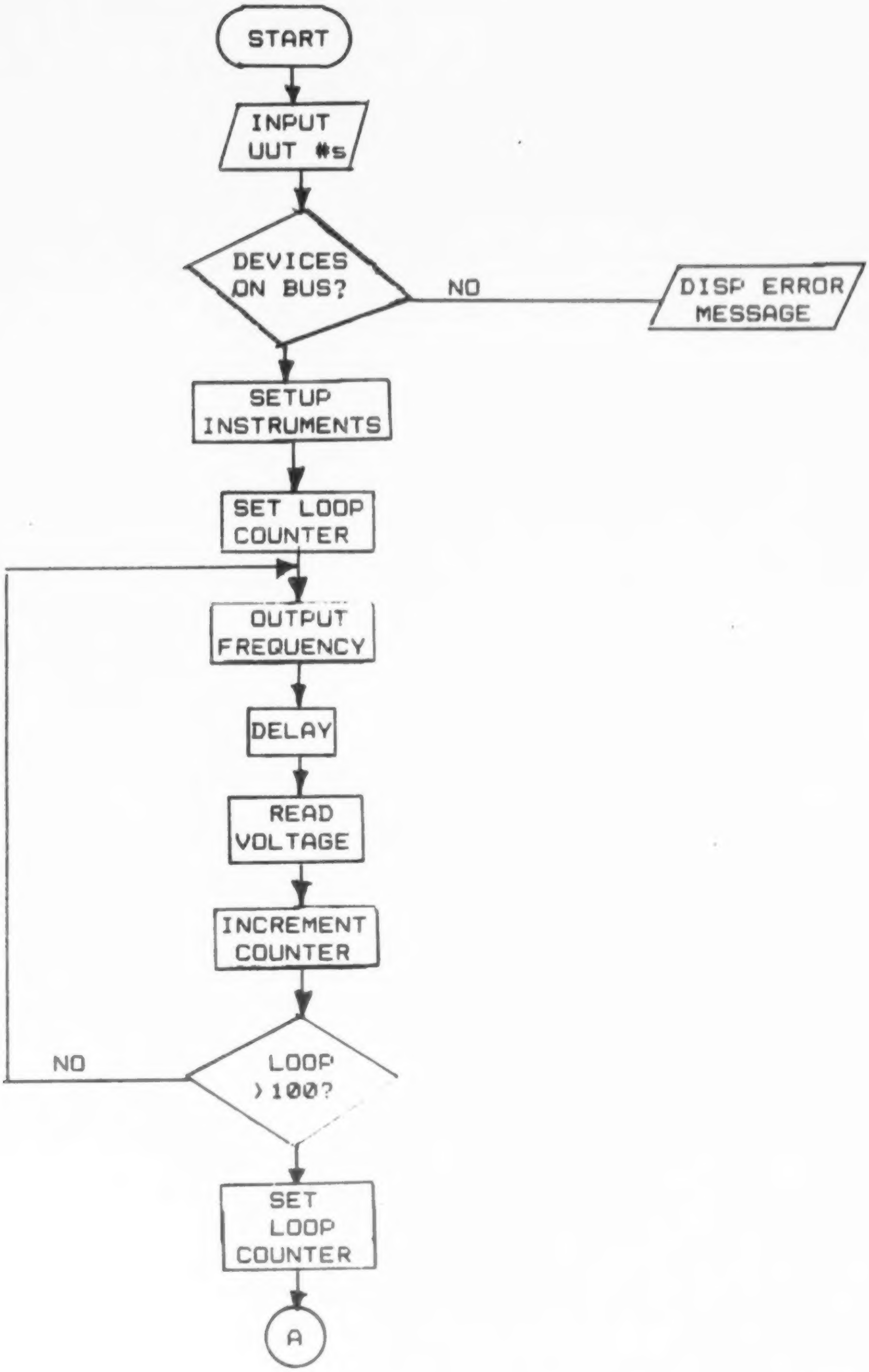
SYMBOLS AND DEFINITIONS:

INPUT/OUTPUT		INPUT FROM OPERATOR - FROM KEYBOARD, SENSING SWITCH ON TEST FIXTURE, DATA FROM A FILE i.e. DATA TO BE USE IN CALCULATIONS.
PROCESS		A TASK TO PERFORM A GIVEN FUNCTION SUCH AS SET UP INSTRUMENTS, TAKE READINGS (NOT STORE IN MASS STORAGE), INCREMENT COUNTER, ETC.
DECISION		A COMPARISON IS MADE AND BRANCHING TO ONE OF TWO OPERATIONS RESULTS. SOME COMMON DECISION TYPES: LOOP COUNTERS, STATUS CHECKS, EQUALITY/INEQUALITY, PROPER INPUT VALUE.
TERMINAL		TO IDENTIFY THE BEGINNING AND ENDING OF A FLOWCHART.
CONNECTOR		USED TO CONNECT FLOWCHARTS WHICH COVER MULTIPLE PAGES AND TO REDUCE LENGTHLY FLOW LINES.
PREDEFINED PROCESS		TO INCORPORATE A LARGE TASK SUCH AS A SUBROUTINE.
FLOW LINES		USED TO SHOW PROGRAM FLOW AND INTERCONNECTIONS, OFTEN CONTAIN ARROWHEADS. NORMAL DIRECTION IS DOWN AND TO THE RIGHT. DARKENED LINES CAN SHOW THE NORMAL FLOW.

NOTE:

FLOWCHARTING EXAMPLE

AMPLIFIER BANDWIDTH TEST

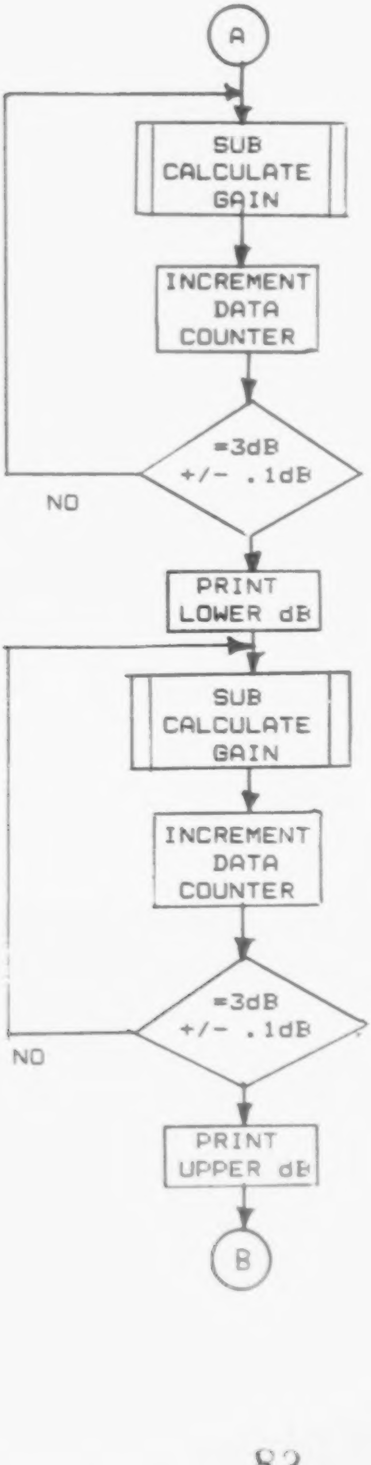

START
INPUT UUT #s
DEVICES ON BUS?
NO
DISP ERROR MESSAGE
SETUP INSTRUMENTS
SET LOOP COUNTER
OUTPUT FREQUENCY
DELAY
READ VOLTAGE
INCREMENT COUNTER
NO
LOOP >100?
SET LOOP COUNTER
A

NOTES:

SOFTWARE: BASIC STRUCTURED PROGRAMMING

FLOWCHARTING EXAMPLE

AMPLIFIER BANDWIDTH TEST

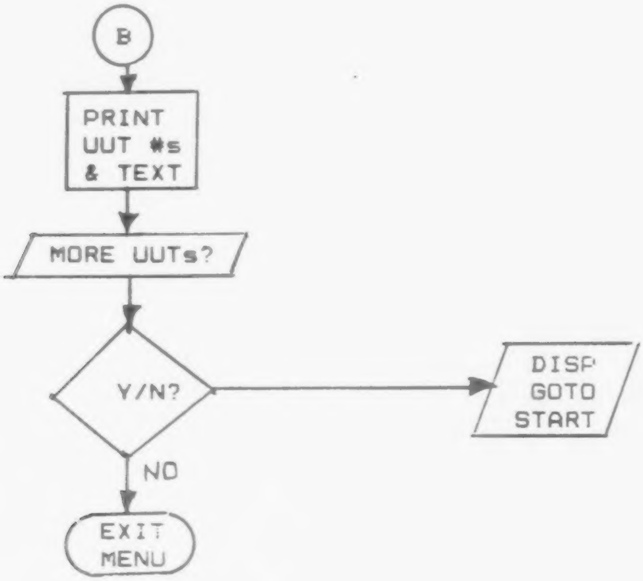

A
SUB
CALCULATE
GAIN
INCREMENT
DATA
COUNTER
=3dB
+/- .1dB
NO
PRINT
LOWER dB
SUB
CALCULATE
GAIN
INCREMENT
DATA
COUNTER
=3dB
+/- .1dB
NO
PRINT
UPPER dB
B

NOTES:

SOFTWARE: BASIC STRUCTURED PROGRAMMING

FLOWCHARTING EXAMPLE

AMPLIFIER BANDWIDTH TEST


B
PRINT UUT #s & TEXT
MORE UUTs?
Y/N?
DISP GOTO START
NO
EXIT MENU

NOTES:

SOFTWARE: BASIC STRUCTURED PROGRAMMING

PSEUDOCODE EXAMPLE

AMPLIFIER BANDWIDTH TEST

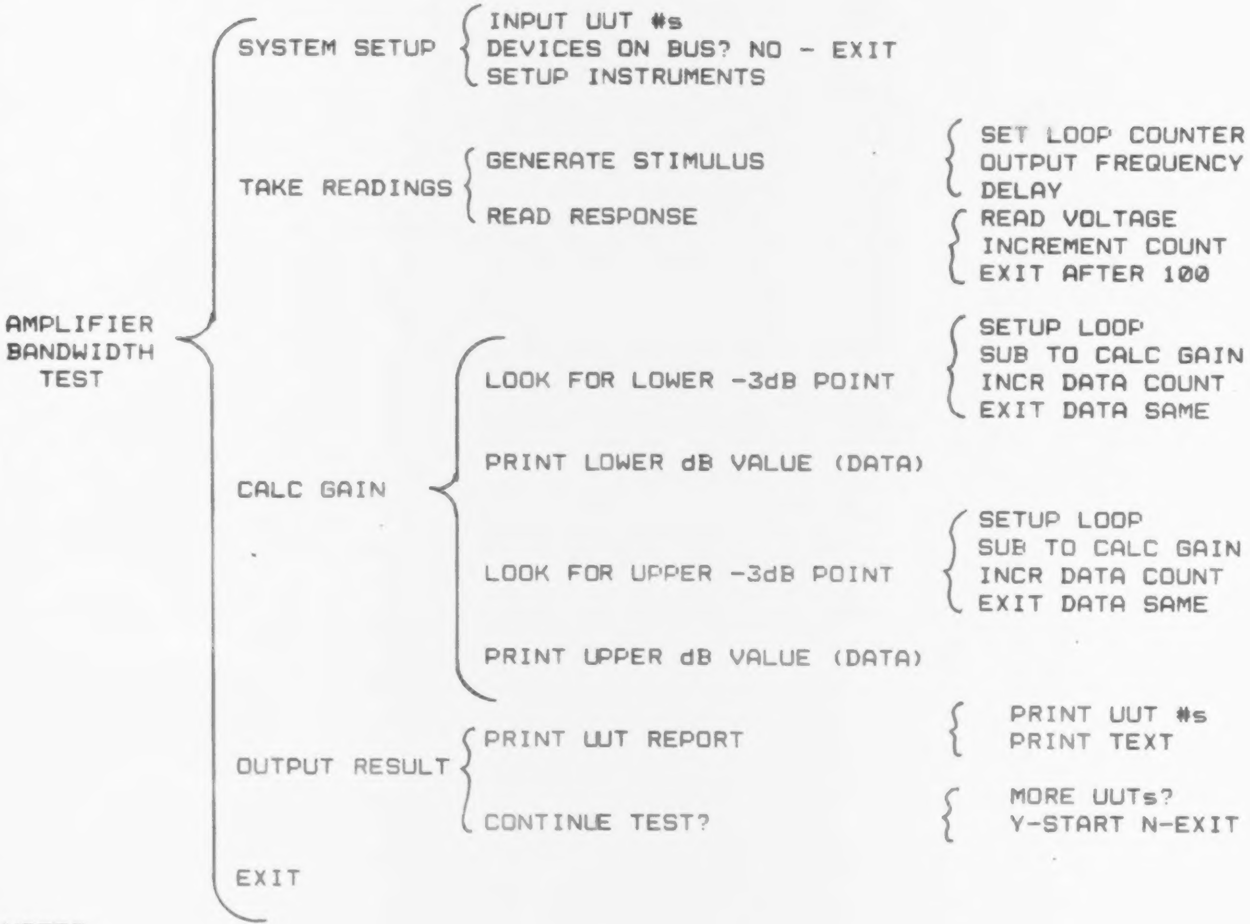
```
        INPUT UUT #s
        CHECK FOR DEVICES ON BUS - IF NOT EXIT
        SETUP INSTRUMENTS
      ( LOOP UNTIL LOOP COUNTER )100
      |    OUTPUT FREQUENCY
      |    DELAY
  *  <     READ VOLTAGE
      |    INCREMENT COUNTER
      |    AFTER 100 LOOPS EXIT
      \ END LOOP
      ( LOOP UNTIL DATA COMPARES
      |    CALL SUB TO CALCULATE GAIN
  *  <     INCREMENT DATA COUNTER
      |    CHECK DATA - IF SAME EXIT
      \ END LOOP
        PRINT LOWER dB VALUE IS (DATA)
      ( LOOP UNTIL DATA COMPARES
      |    CALL SUB TO CALCULATE GAIN
  *  <     INCREMENT DATA COUNTER
      |    CHECK DATA - IF SAME EXIT
      \ END LOOP
        PRINT UPPER dB VALUE IS (DATA)
        PRINT UUT #s & TEXT
        PROMPT OPERATOR - MORE UUTs?
        READ OPERATOR RESPONSE - IF Y DISPLAY "GOTO START", IF N TO EXIT
        EXIT
```

* BRACKETS ARE OPTIONAL, THEY HELP TO FURTHER DEFINE LOOPS.

NOTES:

SOFTWARE: BASIC STRUCTURED PROGRAMMING

WARNIER-ORR EXAMPLE

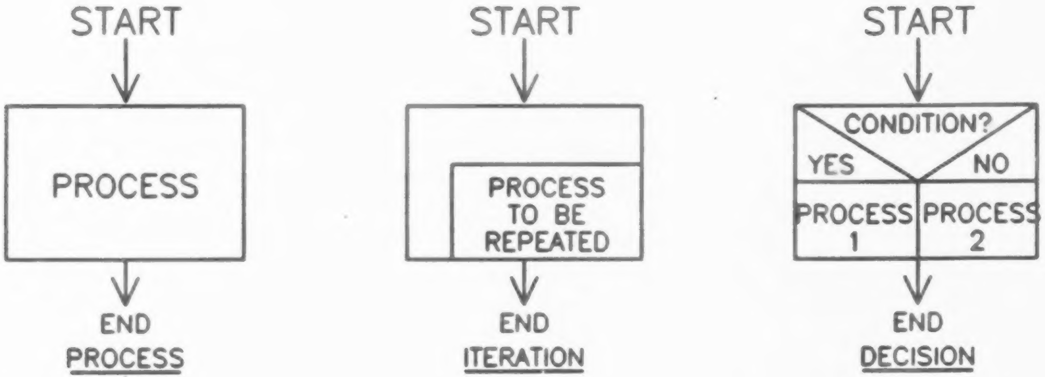


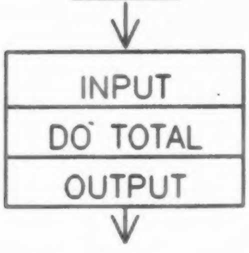
- AMPLIFIER BANDWIDTH TEST
  - SYSTEM SETUP
    - INPUT UUT #s
    - DEVICES ON BUS? NO - EXIT
    - SETUP INSTRUMENTS
  - TAKE READINGS
    - GENERATE STIMULUS
      - SET LOOP COUNTER
      - OUTPUT FREQUENCY
      - DELAY
    - READ RESPONSE
      - READ VOLTAGE
      - INCREMENT COUNT
      - EXIT AFTER 100
  - CALC GAIN
    - LOOK FOR LOWER -3dB POINT
      - SETUP LOOP
      - SUB TO CALC GAIN
      - INCR DATA COUNT
      - EXIT DATA SAME
    - PRINT LOWER dB VALUE (DATA)
    - LOOK FOR UPPER -3dB POINT
      - SETUP LOOP
      - SUB TO CALC GAIN
      - INCR DATA COUNT
      - EXIT DATA SAME
    - PRINT UPPER dB VALUE (DATA)
  - OUTPUT RESULT
    - PRINT UUT REPORT
      - PRINT UUT #s
      - PRINT TEXT
    - CONTINUE TEST?
      - MORE UUTs?
      - Y-START N-EXIT
  - EXIT

NOTES:

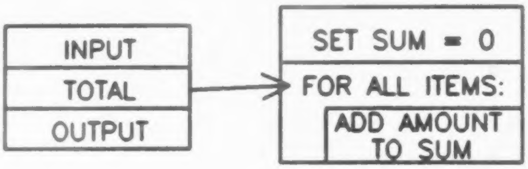
SOFTWARE: BASIC STRUCTURED PROGRAMMING

NASSI-SCHNEIDERMAN DIAGRAM
COURTESY OF HEWLETT PACKARD, REPRINTED WITH PERMISSION

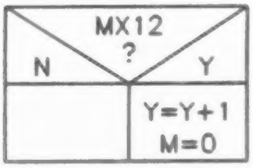

START
PROCESS
END
PROCESS
START
PROCESS
TO BE
REPEATED
END
ITERATION
START
CONDITION?
YES
NO
PROCESS
1
PROCESS
2
END
DECISION


INPUT
DO TOTAL
OUTPUT

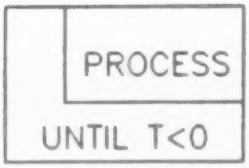
A PROCESS IS OFTEN MADE UP OF SMALLER PROCESSES


INPUT
TOTAL
OUTPUT
SET SUM = 0
FOR ALL ITEMS:
ADD AMOUNT
TO SUM

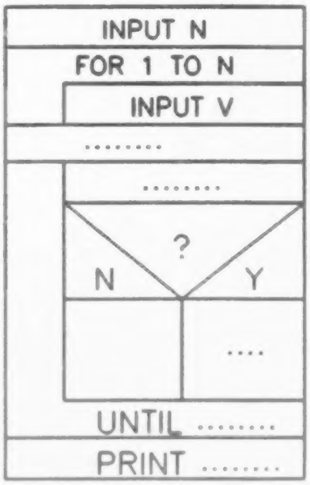
ANY PROCESS MAY BE DRAWN IN FINER DETAIL


MX12
?
N
Y
Y=Y+1
M=0

A DECISION MAY OFTEN HAVE A "BLANK" (UNUSED) ARM


PROCESS
UNTIL T<0

AN ITERATION MAY HAVE THE TERMS OF REPETITION AT START OR END


INPUT N
FOR 1 TO N
INPUT V
....
....
?
N
Y
....
UNTIL ....
PRINT ....

PROGRAM STRUCTURE IS VISIBLE AND NEAT!

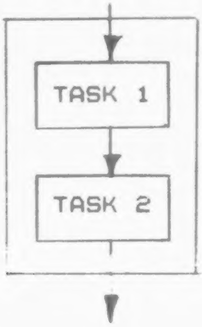
NOTES:

SOFTWARE: BASIC STRUCTURED PROGRAMMING

PROGRAM FLOW

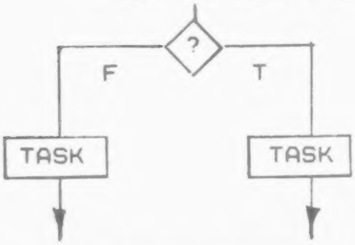
UNCONDITIONAL TASKS

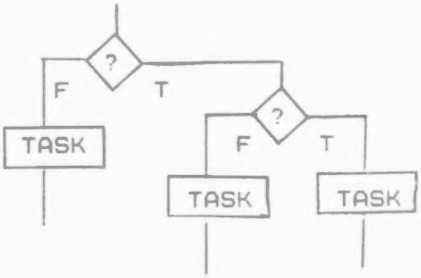
- HAS ONLY ONE ENTRANCE AND EXIT
- MAY CONTAIN SUB-TASKS
- MAY CONTAIN THE FOLLOWING TYPES OF FLOW
  - LINEAR - STRAIGHT THROUGH, NO BRANCHING OR LOOPS
  - BRANCHING - UNCONDITIONAL i.e. "GOTO"
  - REPETITION - LOOPS i.e. "FOR-NEXT"


TASK 1
TASK 2

CONDITIONAL TASKS

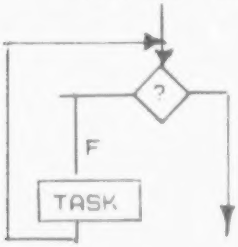
- BASED ON PRIOR SET CONDITION
- MAY BRANCH TO ONE OF MANY SUB TASKS
- MAY BE MULTI-LEVEL

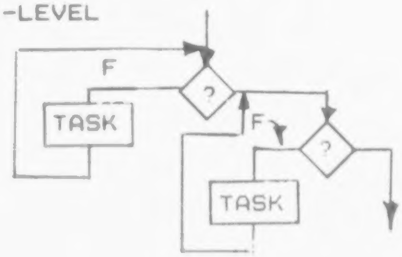

?
F
T
TASK
TASK


?
F
T
TASK
?
F
T
TASK
TASK

REPETITION TASK

- FOR SUB-TASKS REPEATED MORE THAN ONCE
- MAY BE MULTI-LEVEL




F
?
TASK
F
?
TASK

NOTES:

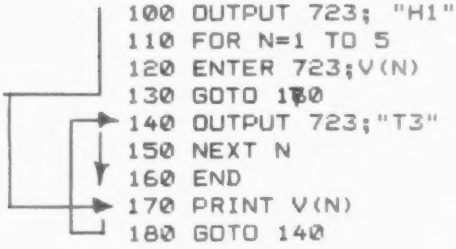
SOFTWARE: BASIC STRUCTURED PROGRAMMING

UNCONDITIONAL BRANCHING

GOTO

THE GOTO SHOULD BE AVOIDED WHEN POSSIBLE BECAUSE IT ALTERS THE STRUCTURE (FLOW) OF THE PROGRAM. THIS MAKES FOLLOWING PROGRAM FLOW DIFFICULT. GOTOs MAY BE NESTED

EXAMPLE:

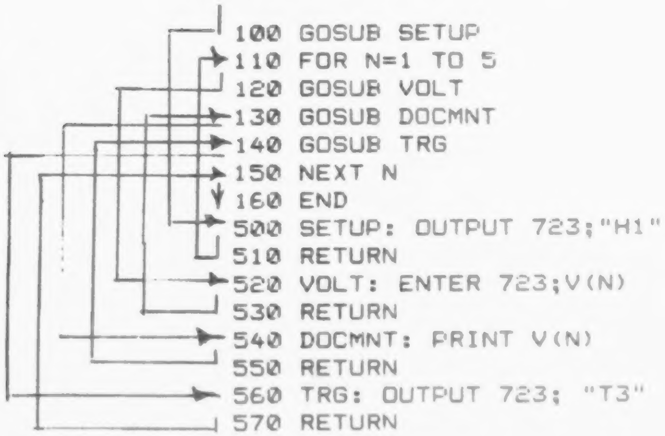


```
100 OUTPUT 723; "H1"
110 FOR N=1 TO 5
120 ENTER 723;V(N)
130 GOTO 170
140 OUTPUT 723;"T3"
150 NEXT N
160 END
170 PRINT V(N)
180 GOTO 140
```

GOSUB

SUBROUTINES ARE RECOMMENDED AS REPLACEMENTS FOR GOTOs, EXCEPT IN SIMPLE APPLICATIONS OR WHERE GOSUBs ARE MORE DIFFICULT TO USE. GOSUBs MAY BE NESTED. FOR EVERY GOSUB THERE MUST BE A RETURN, WHICH IS PLACED IN THE SUBROUTINE. SUBROUTINES CAN HAVE PARAMETERS (VALUES) PASSED TO THEM FROM HIGHER LEVEL TASKS.

EXAMPLE:



```
100 GOSUB SETUP
110 FOR N=1 TO 5
120 GOSUB VOLT
130 GOSUB DOCMNT
140 GOSUB TRG
150 NEXT N
160 END
500 SETUP: OUTPUT 723;"H1"
510 RETURN
520 VOLT: ENTER 723;V(N)
530 RETURN
540 DOCMNT: PRINT V(N)
550 RETURN
560 TRG: OUTPUT 723; "T3"
570 RETURN
```

NOTES:

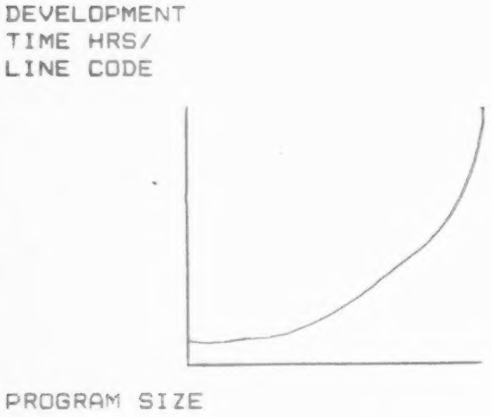
SOFTWARE: BASIC STRUCTURED PROGRAMMING

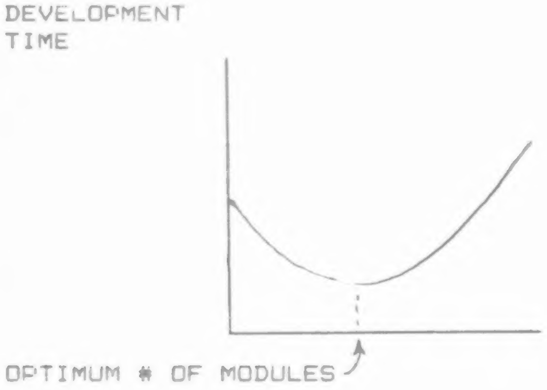
CONCEPTS ADAPTED FROM: WAYNE P. STEVENS, "USING STRUCTURED DESIGN",
JOHN WILEY & SONS PUBLISHERS, NEW YORK, 1981

COMPONENTS OF STRUCTURED DESIGN

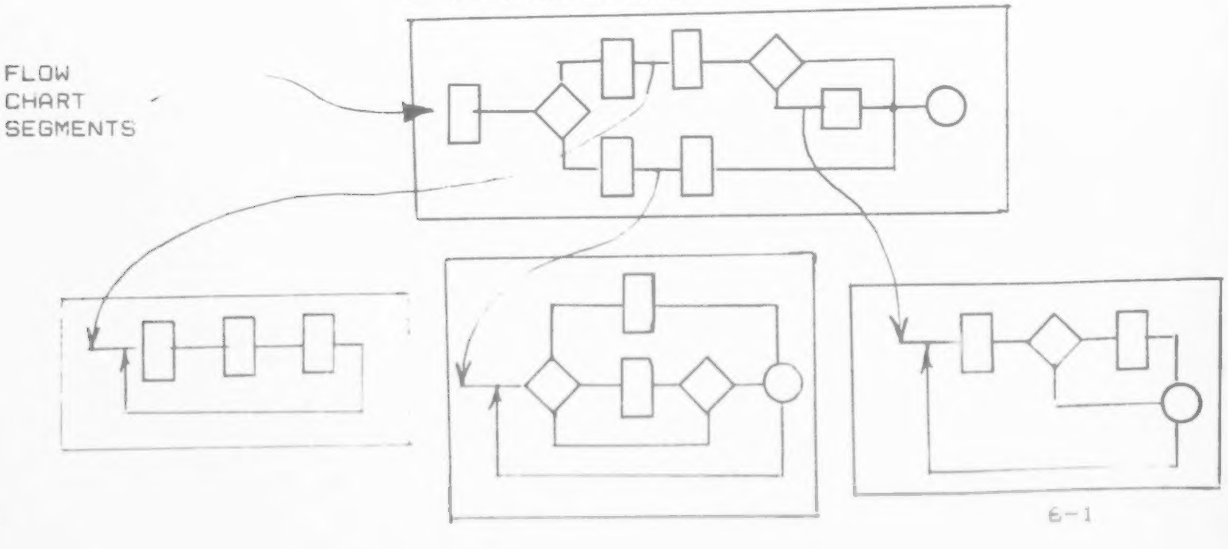
REVIEW OF BENEFITS

- THE PROGRAM STRUCTURE IS SIMPLER
- IT CAN BE WRITTEN, UNDERSTOOD, TESTED, AND CHANGED ONE PIECE AT A TIME
- LESS CHANCE FOR ERRORS WHEN BEING DESIGNED
- SIDE EFFECTS OF PROGRAM MODIFICATION ARE LESS DRAMATIC AND EASIER TO FIND





STRUCTURED CHARTS



NOTES:

SOFTWARE: BASIC STRUCTURED PROGRAMMING

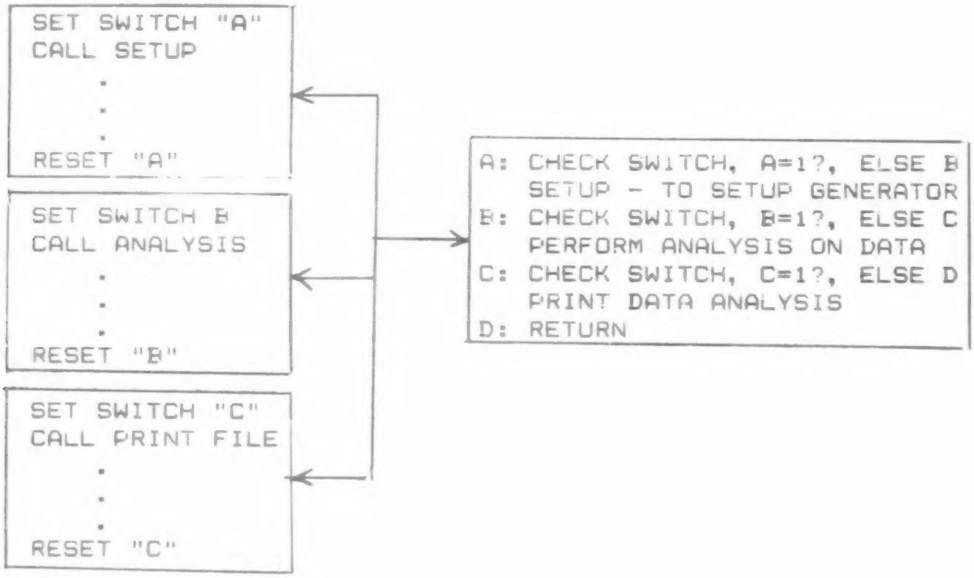
COMPONENTS OF STRUCTURED DESIGN

BINDING

THIS IS A MEASURE ON HOW LINES OF CODE ARE GROUPED (BOUND) TOGETHER. IF THE BINDING LEVEL IS HIGH THEN A MODULE OF CODE SHOULD FUNCTION THE SAME WAY NO MATTER WHAT OTHER PIECE OF SOFTWARE USES IT. THE MAIN OBJECTIVE IS TO INSURE THAT ERRORS IN ONE MODULE WILL NOT CAUSE ANOTHER MODULE TO FAIL. IF THIS WERE TO OCCUR IT COULD BE VERY DIFFICULT TO DETERMINE WHERE THE ERROR IS.

COINCIDENTAL BINDING - LINES OF CODE ARE GROUPED TOGETHER JUST OUT OF COINCIDENCE, MAINLY TO CONSERVE MEMORY SPACE. THIS WOULD BE CODE WHICH IS USED QUITE OFTEN. SOFTWARE "SWITCHES" ARE USED TO SELECT THE PORTION OF COINCIDENTAL CODE NEEDED.

EXAMPLE:



```
SET SWITCH "A"
CALL SETUP
     .
     .
     .
RESET "A"
```

```
SET SWITCH B
CALL ANALYSIS
     .
     .
     .
RESET "B"
```

```
SET SWITCH "C"
CALL PRINT FILE
     .
     .
     .
RESET "C"
```

```
A: CHECK SWITCH, A=1?, ELSE B
   SETUP - TO SETUP GENERATOR
B: CHECK SWITCH, B=1?, ELSE C
   PERFORM ANALYSIS ON DATA
C: CHECK SWITCH, C=1?, ELSE D
   PRINT DATA ANALYSIS
D: RETURN
```

IF A SWITCH IS NOT RESET THEN OTHER MODULES WILL HAVE ERRORS.

NOTES:

COMPONENTS OF STRUCTURED DESIGN

LOGICAL BINDING

CODE WHICH IS LOGICALLY BOUND IS THE TYPE WHICH IS SHARED BY MANY OTHER MODULES, IN THAT IT PERFORMS THE SAME LOGICAL FUNCTION FOR EACH MODULE. FOR EXAMPLE A MODULE TO PRINT A TEST REPORT COULD BE USED BY SEVERAL DIFFERENT TEST PROGRAMS. IT MAY RESIDE ON A HARD DISK AS A SUB-PROGRAM AND BE CALLED (CHAINED) TO THE TEST PROGRAM CALLING IT. WHEN MAKING CHANGES TO THIS MODULE CARE MUST BE TAKEN TO INSURE THAT THE CHANGES ARE "ACCEPTABLE" TO ALL PROGRAMS CALLING IT. THIS TYPE OF BINDING IS WEAK DUE TO THE FACT THAT THE MODULE MUST KNOW THE REPORTING REQUIREMENTS OF THE MODULE ACCESSING IT. CHANGES ARE DIFFICULT TO MAKE.

TEMPORAL BINDING

THIS TYPE OF BINDING RELATES TO CODE WHICH IS TIME RELATED, BUT NOT LOGICALLY RELATED. EXAMPLES OF THIS TYPE OF CODE IS THAT WHICH SCANS INPUTS AND AT THE SAME TIME OUTPUTS INFORMATION TO A DISPLAY. ELEMENTS OF AN INITIALIZATION ROUTINE ARE TEMPORAL BOUND.

COMMUNICATIONAL BINDING

THE ELEMENTS OF THE CODE REFERENCE THE SAME DATA. AN EXAMPLE WOULD BE A MODULE WHICH PRINTS A TEST REPORT ON A BLOCK OF DATA AND WHICH PLOTS BAR CHARTS FROM THAT SAME DATA.

SEQUENTIAL BINDING

THIS IS WHERE THE OUTPUT FROM ONE PROCESS WITHIN THE MODULE IS USED AS THE INPUT DATA TO THE NEXT PROCESS WITHIN THE MODULE. AN EXAMPLE WOULD BE WHERE DATA IS GATHERED INTO A STRING ARRAY, THEN A SUB-STRING ARRAY IS CREATED. IT COULD THEN BE CONVERTED TO ONE OF THREE FORMATS (BCD, BINARY, INTEGER) TO BE USED BY OTHER MODULES.

NOTES:

SOFTWARE: BASIC STRUCTURED PROGRAMMING

COMPONENTS OF STRUCTURED PROGRAMMING

FUNCTIONAL BINDING

THE MODULE IS STRUCTURED TO PERFORM A SPECIFIC TASK. EACH PORTION OF THE MODULE CONTAINS ONLY THE ELEMENTS NECESSARY TO PERFORM ITS PORTION OF THE TASK. EXAMPLE: A PROGRAM TO TEST A BCD COUNTER PRINTED WIRING BOARD (PWB) CONTAINS A MODULE CALLED TEST SEQUENCER WHICH IS COMPOSED OF THE FOLLOWING ELEMENTS: STIMULUS GENERATOR & DATA RECORDER. THE MODULE IS DESIGNED FOR A SPECIFIC TASK AND IS NOT EASILY ADAPTED TO OTHERS. AS ITS ADAPTABILITY IS DIMINISHED, ITS BINDING IS STRENGTHEND i.e. THE MODULE IS VERY APPLICATION DEPENDMENT. IF THE MODULE TEST SEQUENCER WERE BROKEN INTO TWO MODULES RATHER THAN ONE IT WOULD FUNCTION THE SAME WAY.

COUPLING

THIS IS THE MEASURE OF THE STRENGTH OF THE INTERDEPENDENCY BETWEEN MODULES. THE GOAL OF STRUCTURED PROGRAMMING IS TO REDUCE THE COUPLING BETWEEN MODULES. THIS INCREASES THEIR ABILITY TO OPERATE WITH AUTONOMY. AS COUPLING IS INCREASED A GREATER NUMBER OF INTERCONNECTIONS EXIST BETWEEN MODULES. TO REDUCE THIS, GROUP HIGHLY RELATED MODULES TOGETHER . THESE GROUPS CAN BE COUPLED TO OTHER GROUPS WITH FEWER LINES OF COUPLING. THIS ALLOWS THE PROGRAM TO BE EASILY SEGMENTED i.e. BROKEN INTO MAJOR MODULES.

TYPES OF COUPLING

EXTERNAL - THIS IS WHEN A MODULE GAINS ACCESS TO INFORMATION WITHIN ANOTHER MODULE DIRECTLY. TAKE FOR EXAMPLE A SYSTEM THAT WILL HAVE A DMM AT EITHER OF ONE OF TWO ADDRESSES. WITH EXTERNAL COUPLING THE MODULE WHICH NEEDS TO KNOW THE LOCATION OF THE DMM DETERMINES THE ADDRESS ITSELF. THIS MEANS IF THE SYSTEM IS EXPANDED AND ANOTHER MODULE NEEDS THE ADDRESS OF THE DMM, IT WILL HAVE TO GO INTO THE MODULE WHICH DETERMINED THE ADDRESS IN ORDER TO GET IT. THUS IF THE DMM ADDRESSING MODULE HAS NOT BEEN ACCESSED OR IF IT IS MODIFIED IT MAY NOT HAVE THE INFORMATION NEEDED.

PERFORM - THIS TYPE OF MODULE WOULD BE IDENTICAL TO THE PREVIOUS EXAMPLE WITH THE EXCEPTION THAT NO OTHER MODULE WOULD REQUIRE "EXTERNAL" ACCESS TO ITS DATA. THIS TYPE IS STILL UNDESIREABLE BECAUSE THE ADDRESS DETERMINING CODE IS BURIED WITHIN THE MODULE, THUS ITS OPERATION IS NOT AS CLEAR.

CALL - THE MOST DESIREABLE TYPE OF CONNECTION. THIS IS EASY TO FOLLOW, TENDS TO HAVE LESS COUPLING, AND IS LESS ERROR PRONE WHEN CHANGES HAVE TO BE MADE.

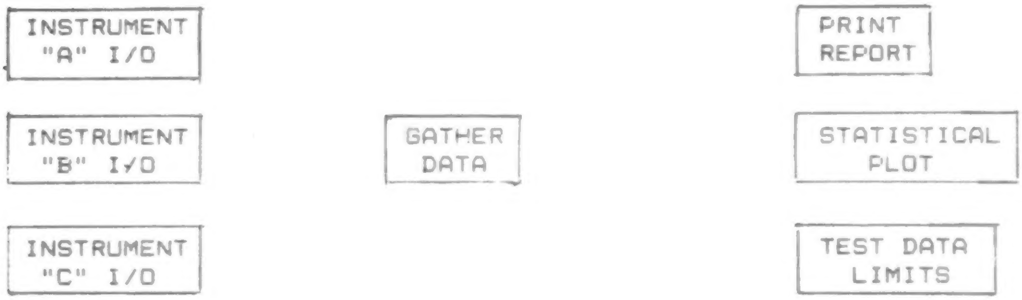
NOTES:

SOFTWARE: BASIC STRUCTURED PROGRAMMING

COMPONENTS OF STRUCTURED PROGRAMMING

SIZE OF CONNECTION

THIS REFERS TO THE AMOUNT OF DISTINCT DATA ITEM TYPES (PARAMETERS) PASSED BY THE INTERFACE EACH TIME IT IS USED. ONE THOUGHT MAY BE TO COMBINE ITEMS UNDER THE SAME PARAMETER TO BE PASSED i.e. WITHIN AN ARRAY. THIS WILL ONLY REDUCE THE CLARITY OF THE OPERATION OF THE INTERFACE. THE SMALLER THE SIZE (LESS DISTINCT DATA TYPES RATHER THAN QUANTITY OF EACH DATA ITEM) THE LESS THE COUPLING. IF A MODULE IS GATHERING DATA FROM SEVERAL INSTRUMENTS AND DISTRIBUTING IT TO SEVERAL PROCESSES IT IS A SHARED DATA SYSTEM WHICH HAS A HIGH DEGREE OF COUPLING. ASSUME THAT THE PROCESSES CAN MODIFY THE DATA AND THE DATA IS ALSO USED TO MODIFY THE I/O OPERATIONS. THIS MEANS THAT THERE ARE LINKS BETWEEN EVERY MODULE FOR EVERY DATA ITEM GATHERED.



THE NUMBER OF LINKS IS $M(M-1) \times D$, WHERE M=NUMBER OF MODULES AND D= NUMBER OF DATA ITEMS. FOR 10 DATA ITEMS: $6(6-1) \times 10 = 300$ POSSIBLE LINKS. THE PROBLEM CAN BE REDUCED IF THE SHARED DATA CAN BE SEGMENTED INTO SMALLER SHARED DATA AREAS.

NOTES:

SOFTWARE: BASIC STRUCTURED PROGRAMMING

COMPONENTS OF STRUCTURED PROGRAMMING

WHAT IS COMMUNICATED - TWO ITEMS MAY BE COMMUNICATED: CONTROL AND DATA.

PASSING CONTROL: ASSUME A MODULE IS TO PERFORM AN ANALYSIS ON DATA PASSED TO IT FROM AN INPUT MODULE. IN ORDER TO PROCESS THE DATA IT MUST BE TOLD WHAT FORMAT THE DATA IS IN FROM A "FORMAT DETERMINING" MODULE. THUS, THE FORMAT DETERMINING MODULE MODIFIES THE OPERATION OF THE ANALYSIS MODULE. A WAY TO ELIMINATE THIS CONTROL FUNCTION IS TO HAVE THE ANALYSIS MODULE CHECK LOCATIONS IN THE INPUT MODULE FOR DATA. THE LOCATION WHERE DATA IS FOUND DETERMINES THE FORMAT i.e. NO FORMAT DETERMINING MODULE IS NEEDED. THIS SOLUTION IS EASIER TO FOLLOW. CONTROL COUPLING REDUCES THE CAPACITY OF MODULES TO OPERATE WITH AUTONOMY. DATA MAY BE PASSED FROM ONE MODULE TO ANOTHER TO DETERMINE WHAT ACTION IS TO BE TAKEN. A PORTION OF AN ADDRESS (AS DATA) PASSED FROM ONE MODULE TO ANOTHER i.e. DMM #1 IS AT 712 AND DATA IS OUTPUT IN BCD, DMM #2 IS AT 718 AND DATA IS OUTPUT IN BINARY. PASSING 12 OR 18 CAN BE USED TO SELECT THE MODE OF PROCESSING. THUS THE ACTUAL DATA DOES NOT HAVE TO BE INTERPERTED.

PASSING DATA: IF DATA IS PASSED AND THE TYPE IS DETERMINED WITHIN THE MODULE RECEIVING IT, THAT MODULE MAY OPERATE WITH AUTONOMY.

NOTES:

SOFTWARE: APPLICATIONS PROGRAMS

FLUKE 8840A DEMONSTRATION PROGRAMS

THE FOLLOWING PROGRAMS WRITTEN FOR THE FLUKE 8840A ARE WRITTEN FOR FOUR DIFFERENT CONTROLLERS (LISTED BELOW). EACH CONTROLLER HAS ITS OWN UNIQUE WAY OF PERFORMING ITS CONTROLLING OPERATIONS. IT IS BEST TO THOROUGHLY REVIEW THE MANUAL(S) OF A CONTROLLER THAT YOU ARE UNFAMILIAR WITH. REMEMBER EVEN IF A LINE OF CODE IS SYNTAX CORRECT, IT MAY NOT FUNCTION AS YOU HAD PLANNED. THE INTENT OF THE FOLLOWING EXAMPLES IS NOT TO TEACH YOU HOW TO PROGRAM WITH A VARIETY OF CONTROLLERS, IT IS TO SHOW YOU SOME EXAMPLES OF REAL WORLD PROGRAMS WHICH HAVE BEEN WRITTEN WITH THE ATRIBUTES WE HAVE STUDIED.

REPRINTED WITH PERMISSION, JOHN FLUKE MFG. CO. INC.

Table 3-3. Immediate-Mode Commands for Various Controllers

FUNCTION PERFORMED	FLUKE-BASIC on 1720A or 1722A	HP-HPL on HP9825 Calculator	HP-BASIC on HP9816-PC and HP-85 Calculator	TEK-BASIC on 4051 Graphics System
INITIALIZE Port -	INIT PORT 0	cli 7	CLEAR 7	INIT
CLEAR Instrument	CLEAR @4	clr 704	CLEAR 704	PRINT @4:"*"
REMOTE Commands	REMOTE @4	rem 704	REMOTE 704	WBYTE @36,17:
LOCAL Control	LOCAL @4	lcl 704	LOCAL 704	WBYTE @36,1:
EXTERNAL TRIGGER	PRINT @4,"T1"	wrt 704,"T1"	OUTPUT 704;"T1"	PRINT @4:"T1"
TRIGGER Instrument	TRIG @4	trg 704	TRIGGER 704	PRINT @4:"?"
GET Output Data	INPUT @4,A	red 704, A	ENTER 704;A (Note 2)	INPUT @4:A
PRINT Data to Screen	PRINT A	prt A	PRINT A (Note 2)	PRINT A
CONFIGURE for VAC	PRINT @4,"F2"	wrt 704,"F2"	OUTPUT 704;"F2"	PRINT @4:"F2"
CONFIGURE for 200Vac	PRINT @4,"R4"	wrt 704,"R4"	OUTPUT 704;"R4"	PRINT @4:"R4"
TRIGGER Continuously	PRINT @4,"T0"	wrt 704,"T0"	OUTPUT 704;"T0"	PRINT @4:"T0"
SUFFIXES Enable	PRINT @4,"Y1"	wrt 704,"Y1"	OUTPUT 704;"Y1"	PRINT @4:"Y1"
GET Data & Suffix	INPUT @4, A,A$	red 704,A,A$ (Note 1)	ENTER 704;A,A$ (Note 2)	INPUT %4:A,A$
PRINT Data & Suffix	PRINT A,A$	prt A$,A	PRINT A,A$ (Note 2)	PRINT A,A$

Notes:

1. Before using A$ on the 9825 is necessary to enter "dim A$[6]" to allocate a string variable. This statement allows six characters.
2. In the HP9816 system, variables cannot be created from the keyboard; they must be created by running a program. (See error 910 for that system.) To get around this, type in a very short program as follows:

```
SCRATCH     (Hit "EXEC" key)
10 A = 0    (Hit "ENTER" key)
20 A$ = ""  (Hit "ENTER" key)
30 END      (Hit "ENTER" key)
(Hit "RUN" key)
```

This program creates the variables 'A' and 'A$' so that they may be accessed in immediate mode and changed at will. This program is not necessary for the HP-85 Calculator.

NOTE

For the examples using the Fluke 1720A or 1722A, the 8840A is plugged into port 0. The port is initialized by the INIT statement, which sends IFC (interface clear).

SOFTWARE: APPLICATIONS PROGRAMS

FLUKE 8840A DEMONSTRATION PROGRAMS

REPRINTED WITH PERMISSION, JOHN FLUKE MFG. CO. INC.

Example Program: Taking 10 Readings

```
1!   This program selects VDC (F1), Autorange(RO), Slow rate(SO), continuous
2!   trigger(TO), Suffixes enabled (Y1). The program takes 10 readings and
3!   stops. The 8840A is addressed to port 0 device #4 under control of a
4!   Fluke 1720A or 1722A Instrument Controller. In the INPUT statement, the
5!   controller assigns the first part of the 8840A output response (the
6!   measurement value) to R, and assigns the second part (the suffix) to R$.
7!
18!==================FLUKE-BASIC on 1720A, 1722A===============================
19!
20   INIT PORT 0                                     !Clear port
25   CLEAR PORT 0                                    !Clear instr to default functions
35   PRINT @4, "F1 RO SO TO Y1"                      !Write functions to instrument
40     FOR I = 1 TO 10
45        INPUT @4,R,R$                              !Get data from 8840A
50        PRINT "      ",I,R,R$                      !Print to 1722 display
55     NEXT I
60   END
```

```
17!        EQUIVALENT PROGRAM FOR HP9816 PC AND HP-85 CALCULATOR IN HP-BASIC
18!===========================================================================
19!
20   CLEAR 7                                         !Clear port
25   CLEAR 704                                       !Clear instr to default functions
35   OUTPUT 704;  "F1 RO SO TO Y1"                   !Write functions to instrument
40     FOR I = 1 TO 10
45        ENTER 704; R,R$                            !Get data from 8840A
50        PRINT "      ",I,R,R$                      !Print to HP9816 display
50        PRINT I;R;R$                    !Substitute this PRINT statement for HP-85
55     NEXT I
60   END
```

```
!                      EQUIVALENT PROGRAM FOR HP9825 CALCULATOR
!=====================================================================
!
 0:  dim R$[6]                               !Allocate string variable up to 6 characters
 1:  fxd 5                                   !Set display to 5 decimal places
 2:  cli 7                                   !Clear port
 3:  clr 704                                 !Clear instr to default functions
 4:  10 -> I                                 !Initialize reading counter
 5:  wrt 704,  "F1 RO SO TO Y1"              !Write functions to instrument
 6:  "IN":  I-1 -> I                         !Decrement reading counter
 7:         red 704, R,R$                    !Get data from 8840A
 8:         prt R$,R                         !Print to HP9825 printer
 9:   if I<>0;  gto "IN"
10:  end
```

```
17REM  EQUIVALENT PROGRAM FOR TEKTRONIX 4051 SERIES GRAPHICS SYSTEM IN TEK-BASIC
18REM===========================================================================
19REM
20   INIT                                            REM Clear port
35   PRINT @4: "F1 RO SO TO Y1"                      REM Write functions to instrument
40     FOR I = 1 TO 10
45        INPUT %4:R,R$                              REM Get data from 8840A
50        PRINT "      ",I,R,R$                      REM Print to 1722 display
55     NEXT I
60   END
```

SOFTWARE: APPLICATIONS PROGRAMS

FLUKE 8840A DEMONSTRATION PROGRAMS

REPRINTED WITH PERMISSION, JOHN FLUKE MFG. CO. INC.

Example Program: Taking Readings with Local Control

```
1! This is a sample program which commands the 8840A to the default state of
2! VDC, autorange, slow rate, continuous trigger.  All readings appear simul-
3! taneously on the instrument display while being logged on the controller
4! display with suffixes enabled for function readout. Full local control is
5! given to the 8840A so that any range or function may invoked easily. The
6! controller always echos the 8840A display while this program is running.
7! The first copy of this program runs on Fluke 1720A and 1722A Controllers.
8! The second runs on the HP9816 personal computer running HP-BASIC. The third
9! runs on the HP9825 (HPL) calculator.
10'
11'===============FLUKE BASIC on 1720A, 1722A=====================================
12'
20'=================================INITIALIZATION
30 INIT PORT 0                      !Clear port
40 CLEAR PORT 0                     !Clear Instrument
50 WAIT 1000                        !Wait 1 second before sending command
60 PRINT @4,"F1 R0 S0 T0 Y1"        !F1=VAC, R0=autorange, S0=slow rate
70'                                 !T0=continuous trigger, Y1=enable suffix
80 X=0
90 LOCAL @4                         !Give local control to instrument
95'
100'=================================MAIN
110 INPUT @4,A,A$                     !Get reading and suffix
120   X=X+1                           !Increment reading count
130   PRINT "            ",X,A,A$     !Display readings incl suffix
140 GOTO 110
145'
200 END
```

```
11'       EQUIVALENT PROGRAM FOR HP9816 PC AND HP-85 CALCULATOR IN HP-BASIC
12'==============================================================================

20'=================================INITIALIZATION
30 CLEAR 7                          !Clear HPIB port
40 CLEAR 704                        !Clear instrument
60 OUTPUT 704;"F1 R0 S0 T0 Y1"      !F1=VAC, R0=autorange, S0=slow rate
70'                                 !T0=continuous trigger, Y1=enable suffix
80 X=0
90 LOCAL 704                        !Give local control to instrument
95'
100'=================================MAIN
110 ENTER 704;A,A$                    !Get reading and suffix
120   X=X+1                           !Increment reading count
130   PRINT "            ",X,A,A$     !Display readings incl suffix (Use for HP9816
130   DISP  X;A;A$     !For HP-85, substitute this line for PRINT statement above
140 GOTO 110
145'
200 END
```

```
'                  EQUIVALENT PROGRAM FOR HP9825 CALCULATOR
'===========================================================================
'
'================================INITIALIZATION
 0:  dim A$[6]                      !Allocate string variable up to 6 characters
 1:  fxd 5                          !Set display to 5 decimal places
 2:  cli 7                          !Clear HPIB port
 3:  clr 704                        !Clear instrument
 4:  wait 1000
 5:  wrt 704, "F1 R0 S0 T0 Y1"      !F1=VAC, R0=autorange, S0=slow rate
'                                   !T0=continuous trigger, Y1=enable suffix
 6:  0 -> X
 7:  lcl 704                        !Give local control to instrument

'================================MAIN
 8:  "IN": red 704,A,A$             !Get reading and suffix
 9:  X+1 -> X                       !Increment reading count
10:  dsp X,A,A$                     !Display readings incl suffix
11:  gto "IN"

12:  end
```

```
10REM  EQUIVALENT PROGRAM FOR TEKTRONIX 4051 SERIES GRAPHICS SYSTEM IN TEK-BASIC
11REM==========================================================================
12REM
20REM===============================INITIALIZATION
30 INIT                            REM Clear port
60 PRINT @4: "F1 R0 S0 T0 Y1"      REM F1=VAC, R0=autorange, S0=slow rate
70REM                              REM T0=continuous trigger, Y1=enable suffix
80 X=0
90 WBYTE @36,1:                    REM Give local control to instrument
95REM
100REM=============================MAIN
110 INPUT %4:A,A$                          REM Get reading and suffix
120    X=X+1                               REM Increment reading count
130    PRINT "             ",X,A,A$        REM Display readings incl suffix
140 GOTO 110
145REM
200 END
```

SOFTWARE: APPLICATIONS PROGRAMS

FLUKE 8840A DEMONSTRATION PROGRAMS

REPRINTED WITH PERMISSION, JOHN FLUKE MFG. CO. INC.

Example Program: Using the Serial Poll Register

```
1! The following program illustrates one possible use of the serial poll
2! register.  In this case it is merely looking for data available.  The in-
3! strument is addressed to port 0 device #4 under control of a FLUKE 1720A
4! 1720A or 1722A controller. The function of the program is to display on the
5! 1720A screen the lowest resistance seen across the input terminals of the
6! 8840A using the 2-wire ohms function in autorange.  See function list for
7! interpretation of line 110.
8!
98!===============FLUKE BASIC on 1720A or 1722A================================
99!
100 INIT PORT 0  \  CLEAR PORT 0  \  WAIT 500   !Clear port,clear inst, delay
110 PRINT @4, "N3001P0 Y1 ?"                   !Instr functions + trigger
120 INPUT @4,A,A$                               !Get first reading
130 PRINT "      ",A,A$;"S (2-wire) LOWEST READING"
140 TRIG @4
160 B%=SPL(4)                                   !get Serial Poll Reg @ addr #1
170  IF (B%=0%)    THEN 160                     !Looking for data available
175    INPUT @4,R,A$                            !Get next data
180  IF R )=A  THEN 140                         !Throw away data if not lowest
185    A=R                                      !Update lowest reading
190    GOTO 130                                 !Print new low
900 END

97!        EQUIVALENT PROGRAM FOR HP9816 PC AND HP-85 CALCULATOR IN HP-BASIC
98!==============================================================================
99!
100 CLEAR 7                                     !Clear HPIB port
101 CLEAR 704                                   !Clear 8840A
110 OUTPUT 704; "N3001P0 Y1 ?"                  !Instr functions + trigger
120 ENTER 704; A,A$                             !Get first reading
130 PRINT "      ",A,A$;"S (2-wire) LOWEST READING"    !Use this line for HP9816
130 DISP A;A$;"S LOWEST READ" !For HP-85, substitute this for PRINT statement above
140 TRIGGER 704
160 B=SPOLL(704)                                !get Serial Poll Reg @ addr #1
170  IF (B=0)   THEN 160                        !Looking for data available
175    ENTER 704;R,A$                           !Get next data
180  IF R )=A  THEN 140                         !Throw away data if not lowest
185    A=R                                      !Udata lowest reading
190    GOTO 130                                 !Print new low
900 END

!              EQUIVALENT PROGRAM FOR HP9825 CALCULATOR
!===========================================================================
!
 0:  dim A$[6]                              !Allocate string variable up to 6
                                            characters
 1:  fxd 3                                  !Fix 3 decimal places
 2:  cli 7                                  !Clear HPIB port
 3:  clr 704                                !clear inst
 4:  wrt 704, "N3001P0 Y1 ?"                !Instr functions + trigger
 5:  red 704, A,A$                          !Get first reading
 6:  "DISP": dsp A,A$,"-2w LOWEST READ"
 7:  "TRIG": trg 704
 8:  "POLL": rds (704) -) X                 !get Serial Poll Reg @ addr #1
 9:   if (B=0);  gto "POLL"                 !Looking for data available
10:     red 704,R,A$                        !Get next data
11:   if R)=A;  gto "TRIG"                  !Throw away data if not lowest
12:     R -) A;   gto "DISP"                !Udata lowest reading
13:  END


97REM  EQUIVALENT PROGRAM FOR TEKTRONIX 4051 SERIES GRAPHICS SYSTEM IN TEK-BASIC
98REM=============================================================================
99REM
100 INIT                                       REM Clear port
110 PRINT @4: "N3001P0 Y1 ?"                   REM Instr functions + trigger
120 INPUT %4:A,A$                              REM Get first reading
130 PRINT "      ",A,A$;"S (2-wire) LOWEST READING"
140 PRINT @4: "?"                              REM Trigger reading
160 WBYTE @95,24:                              REM Turn on serial-poll-enable
165 WBYTE @68:                                 REM At address #4
167 RBYTE B                                    REM Get serial poll register
168 WBYTE @95,25:                              REM Turn off serial-poll-enable
170  IF (B=0)     THEN 160                     REM Looking for data available
175    INPUT %4:R,A$                           REM Get next data
180  IF R )=A  THEN 140                        REM Throw away data if not lowest
185    A=R                                     REM Update lowest reading
190    GOTO 130                                REM Print new low
900 END
```

SOFTWARE: APPLICATIONS PROGRAMS

FLUKE 8840A DEMONSTRATION PROGRAMS

REPRINTED WITH PERMISSION, JOHN FLUKE MFG. CO. INC.

Example Program: Interrupt Processing

```
1! The following program illustrates one possible use of the serial poll
2! register.  In this case it is merely looking for data available.  The in-
3! strument is addressed to port 0 device #4 under control of a FLUKE 1720A
4! 1720A or 1722A controller. The function of the program is to display on the
5! 1720A screen the lowest resistance seen across the input terminals of the
6! 8840A using the 2-wire ohms function in autorange.  See function list for
7! interpretation of line 110.
8!
98!===============FLUKE BASIC on 1720A or 1722A=============================
99!
100 INIT PORT 0  \  CLEAR PORT 0  \  WAIT 500  !Clear port,clear inst, delay
110 PRINT @4, "N3001P0 Y1 ?"                   !Instr functions + trigger
120 INPUT @4,A,A$                              !Get first reading
130 PRINT "     ",A,A$;"S (2-wire) LOWEST READING"
140 TRIG @4
160 B%=SPL(4)                                  !get Serial Poll Reg @ addr #1
170  IF (B%=0%)    THEN 160                    !Looking for data available
175    INPUT @4,R,A$                           !Get next data
180  IF R>=A  THEN 140                         !Throw away data if not lowest
185    A=R                                     !Update lowest reading
190    GOTO 130                                !Print new low
900 END
```

```
97!        EQUIVALENT PROGRAM FOR HP9816 PC AND HP-85 CALCULATOR IN HP-BASIC
98!======================================================================
99!
100 CLEAR 7                                    !Clear HPIB port
101 CLEAR 704                                  !Clear 8840A
110 OUTPUT 704; "N3001P0 Y1 ?"                 !Instr functions + trigger
120 ENTER 704; A,A$                            !Get first reading
130 PRINT "     ",A,A$;"S (2-wire) LOWEST READING"     !Use this line for HP9816
130 DISP A;A$;"S LOWEST READ"      !For HP-85, substitute this for PRINT statement above
140 TRIGGER 704
160 B=SPOLL(704)                               !get Serial Poll Reg @ addr #1
170  IF (B=0)    THEN 160                      !Looking for data available
175    ENTER 704;R,A$                          !Get next data
180  IF R>=A  THEN 140                         !Throw away data if not lowest
185    A=R                                     !Udata lowest reading
190    GOTO 130                                !Print new low
900 END
```

```
!                EQUIVALENT PROGRAM FOR HP9825 CALCULATOR
!=================================================================
!
 0: dim A$[6]                                 !Allocate string variable up to 6 characters
 1: fxd 3                                     !Fix 3 decimal places
 2: cli 7                                     !Clear HPIB port
 3: clr 704                                   !clear inst
 4: wrt 704, "N3001P0 Y1 ?"                   !Instr functions + trigger
 5: red 704, A,A$                             !Get first reading
 6: "DISP": dsp A,A$,"-2w LOWEST READ"
 7: "TRIG": trg 704
 8: "POLL": rds (704) -> X                    !get Serial Poll Reg @ addr #1
 9:  if (B=0);  gto "POLL"                    !Looking for data available
10:    red 704,R,A$                           !Get next data
11:  if R>=A;  gto "TRIG"                     !Throw away data if not lowest
12:    R -> A;    gto "DISP"                  !Udata lowest reading
13: END
```

```
97REM  EQUIVALENT PROGRAM FOR TEKTRONIX 4051 SERIES GRAPHICS SYSTEM IN TEK-BASIC
98REM=====================================================================
99REM
100 INIT                                       REM Clear port
110 PRINT @4: "N3001P0 Y1 ?"                   REM Instr functions + trigger
120 INPUT %4:A,A$                              REM Get first reading
130 PRINT "     ",A,A$;"S (2-wire) LOWEST READING"
140 PRINT @4: "?"                              REM Trigger reading
160 WBYTE @95,24:                              REM Turn on serial-poll-enable
165 WBYTE @68:                                 REM At address #4
167 RBYTE B                                    REM Get serial poll register
168 WBYTE @95,25:                              REM Turn off serial-poll-enable
170  IF (B=0)      THEN 160                    REM Looking for data available
175    INPUT %4:R,A$                           REM Get next data
180  IF R>=A  THEN 140                         REM Throw away data if not lowest
185    A=R                                     REM Update lowest reading
190    GOTO 130                                REM Print new low
900 END
```

FLUKE 8840A DEMONSTRATION PROGRAMS

REPRINTED WITH PERMISSION, JOHN FLUKE MFG. CO. INC.

Example Program: Interrupt Processing (cont)

```
!                  EQUIVALENT PROGRAM FOR HP9825 CALCULATOR
!======================================================================

!===================INITIALIZE=====================================
  0: dim A$[6]                 !Allocate string variable up to 6 characters
  1: flt 5                     !Set display to 5 places
  2: cli 7                     !Clear HPIB port
  3: clr 704                   !Clear instrument
  4: wrt 704, "N4P1 Y1"
  5: lcl 704
  6: oni 7, "SRG"              !Direct interrupts on chan 7 to instr 500
  7: eir  7                    !Enable interrupts on HPIB (SRG)

!===================MAIN==========================================
  8:"HERE":  gto  "HERE"       !Simulated Controller activity with other matters

!===================SERIAL POLL===================================
  9:"SRG": rds (704) -> X      !This code is looking for the interrupting device.
 10:      if X()0; gto "DVM"   !Since there is only one device of interest, lines
!   rds (705) -> X             !530 and 540 are commented out but illustrated one
!       if X()0; gto "CAL"     !possible way to implement

!===================INTERRUPT ROUTINE FOR 8840A======================
 11:"DVM": red 704, A,A$       !Get data from 8840A
 12:         prt A$,A          !Print data to HP9825
 13: eir 7                     !Re-enable interrupts on HPIB
 14: iret

 15: end
```

```
98REM EQUIVALENT PROGRAM FOR TEKTRONIX 4051 SERIES GRAPHICS SYSTEM IN TEK-BASIC
99REM=========================================================================
100REM=========================INITIALIZE=====================================
110 INIT                                     REM Clear port
120 PRINT @4: "N4P1 Y1"                      REM SRG mask (interrupt only on SRG)
130 WBYTE @36,1:                             REM Set instrument to LOCAL control
140 ON SRG THEN 500

400REM==========================MAIN========================================
410 GOTO 410          REM This is, of course, a locked loop but it represents
420REM                REM an arbitrary amount of time when the controller
430REM                REM can be doing other things while waiting for readings
440REM                REM from port 0 on an INTERRUPT basis.

500REM=========================SERIAL POLL===================================
510 POLL P1,P2;4;5;6                 REM This statement is generalized to
520 GOTO P1 OF 600,700,800           REM include devices 4, 5, 6 but of course
530REM                               REM 5 and 6 do not exist in this example.
540REM                               REM Parallel poll is particularly
550REM                               REM convenient in this controller.

600REM=========================INTERRUPT ROUTINE for 8840A====================
610 INPUT %4:A,A$                    REM Get data.
620 PRINT "     ",A,A$               REM Print to 4051 screen.
630 RETURN                           REM Resume previous task. Return
640REM                               REM address is memorized automatically
650REM                               REM just like the "ON ERROR" mechanism
660REM                               REM in BASIC (in this case line 410).

900 END
```

```
@PLm>1!     THIS SOFTWARE IS FOR DEMONSTATION PURPOSES ONLY.  It is, therefore,
2!      intended to be informational in nature only.  Implementation in a
3!      user's application should only be made after careful analysis by
4!      the user's own software experts.  John Fluke Mfg. Co. Inc.
5!      specifically discalims all warranties on such software, express
6!      or implied, including but not limited to any warranty of
7!      merchantability, fitness, or adaquacy for any particular purpose
8!      or use.
```

REPRINTED WITH PERMISSION, JOHN FLUKE MFG. CO. INC.

```
9!
10 REM ******************** DISPLAY CONTROLS ****************************
20 E$ = CHR$(27) +"["                          ! escape for enhancements
30 REM ***************** ENHANCEMENTS ********************************
40 EO$ = E$ + "m"                               ! enhancements off
50 HI$ = E$ + "1m"                              ! high intensity
60 UL$ = E$ + "4m"                              ! underline
70 BL$ = E$ + "5m"                              ! blinking
80 RI$ = E$ + "7m"                              ! reverse image
90 NS$ = E$ + "p"                              ! normal size
100 DS$ = E$ + "1p"                             ! double size
110 GF$ = E$ + "2p"                             ! graphics off
120 GN$ = E$ + "3p"                             ! graphics on
130 KE$ = E$ + "4p"                             ! enable keyboard
140 KD$ = E$ + "5p"                             ! disable keyboard
150 REM ****************** ERASING ***********************************
160 LE$ = E$ + "K"                              ! to line end
170 LS$ = E$ + "1K"                             ! from line start
180 AL$ = E$ + "2K"                             ! all of line
190 DE$ = E$ + "J"                              ! to display end
200 DB$ = E$ + "1J"                             ! from display beginning
210 AD$ = E$ + "2J"                             ! all of display
220 BP$ = CHR$(7%)                              ! beep
230 REM ********************* ARRAYS **********************************
240 DIM CF$(9%)                       ! CAL FUNCTION
250 CF$(1%) = "PERFORM COMPLETE CALIBRATION?"
260 CF$(2%) = "[ ] PERFORM DC CALIBRATION?"
270 CF$(3%) = "[ ] PERFORM AC CALIBRATION?"
280 CF$(4%) = "[ ] PERFORM OHMS CALIBRATION?"
290 CF$(5%) = "[ ] PERFORM DC CURRENT CALIBRATION?"
300 CF$(6%) = "[ ] PERFORM AC CURRENT CALIBRATION?"
310 CF$(7%) = "[ ] EXIT"
320 CF$(8%) = "BEGIN" \ CF$(9%) = "CANCEL"
330 Y% = 0% \ Y1% = 0% \ Y2% = 0% \ Y3% = 0% \ Y4% = 0%
340 Z% = 0% \ AD% = 0% \ AC% = 1%
350 REM **************** SET-UP INSTRUCTIONS ************************
360 INIT PORT 0% \ CLEAR PORT 0% \ REMOTE PORT 0
370 PRINT EO$; NS$; AD$;                          ! CLEAR DISPLAY
380 PRINT CPOS(2%,5%); "Use this program to Software Calibrate the 8840A."
390 PRINT CPOS(3%,5%); "It uses a 5100B as the source and a 8506A to enhance"
400 PRINT CPOS(4%,5%); "the accuracy of the 5100B output for calibration."
410 PRINT CPOS(6%,5%); "Set the instrument address switches as follows:"
420 PRINT CPOS(7%,4%); HI$;"5100B address switch to 1"
430 PRINT CPOS(8%,5%); "8506A address switch to 2"
440 PRINT CPOS(9%,5%); "8840A address switch to 3";EO$
450 PRINT CPOS(16%,55%); EO$;
460 PRINT CPOS(16%,30%); RI$; "Touch SCREEN to continue"; EO$;
470 K% = KEY \ WAIT FOR KEY \ K%=KEY
480 REMOTE @1% @2% @3%
490 Y% = SPL(2%) \ Y% = SPL(3%)
500 GOSUB 9000                                    ! PERFORM SELF TEST
510 ON SRQ GOTO 8000
```

```
520 PRINT @3%, "N40P1"
530 PRINT @1%, "0V,S"
540 PRINT E0$; NS$; AD$;                        ! CLEAR DISPLAY
550 PRINT CPOS(3%,5%); "Enable the 8840A Calibration Mode by pushing"
560 PRINT CPOS(4%,5%); "the Calibration Switch in."
570 PRINT CPOS(16%,55%); E0$;
580 PRINT CPOS(16%,30%); RI$; "Touch SCREEN to continue"; E0$;
590 K% = KEY \ WAIT FOR KEY \ GOTO 30000
1000 REM ********************** DC CAL ****************************
1010 PRINT AD$; CPOS(2%,7%); "DC CALIBRATION"
1020 PRINT CPOS(3%,7%); "PERFORM A/D CALIBRATION"
1030 PRINT CPOS(7%,11%);E0$; CPOS(7%,5%); RI$; "YES";E0$;
1040 PRINT CPOS(7%,36%); E0$; CPOS(7%,32%); RI$; "NO "; E0$;
1050 K% = KEY \ WAIT FOR KEY \ K% = KEY
1060 IF K% = 51% GOTO 1090
1070 IF K% = 60% THEN GOTO 1380 ELSE 1050
1080 ! &&&&&&&&&&&&&&&&&&&& A/D CALIBRATION &&&&&&&&&&&&&&&&&&&&&&&&&&
1090 AD% = 1%
1100 PRINT CPOS(3%,7%); DE$; BL$; "PERFORMING A/D CALIBRATION"; E0$
1110  FOR T% = 1% TO 2%
1120    PRINT @3%, "C1"
1130    WAIT FOR SRQ
1140    GOSUB 6030
1150    IF T% = 1% THEN PRINT CPOS(7%,6%); E0$; LE$; "FIRST ITERATION"
1160    IF T% = 2% THEN PRINT CPOS(7%,6%); E0$; LE$; "SECOND ITERATION"
1170    PRINT CPOS(8%,1%); DE$;
1180    PRINT @2%,"VRF20,"
1190      FOR I% = 1% TO 11%
1200        PRINT @3%, "G2"
1210        INPUT @3%, CP
1220        PRINT CPOS(5%,6%); LE$; HI$; CP; "V CALIBRATION" ;E0$;
1230        PRINT @1%, CP;"V,N" \ WAIT 3000
1240        PRINT @2%, "?" \ INPUT @2%, R
1250        IF R<>CP THEN GOSUB 1320
1260        PRINT @3%, "C0"
1270        WAIT FOR SRQ
1280      NEXT I%
1290    PRINT @1%, "0V, N"
1300  NEXT T%
1310 GOTO 1380
1320 ! &&&&&&&&&&&&&&&&& ADJUST OUTPUT OF 5100B &&&&&&&&&&&&&&&&&&&&&&
1330 PRINT @2%, "?" \ INPUT @2%, R
1340 A1 = CP - R \ PRINT @1, CP + A1; "V,N" \ WAIT 3000
1350 PRINT @2%, "?" \ INPUT @2%, R
1360 IF CP < R - 1E-5 OR CP > R + 1E-5 THEN GOTO 1330 ELSE 1370
1370 RETURN
1380 ! &&&&&&&&&&&&&&&&&&&&&& GAIN CAL &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1390 PRINT @2%,"VROF20,"
1400 PRINT @3%, "F1"
1410 PRINT CPOS(3%,7%); DE$; BL$; "PERFORMING RANGE CALIBRATION"; E0$
1420 GOSUB 6310
1430   FOR I% = 1% TO 5%
1440     PRINT @3%, "G2"
1450     INPUT @3%, CP
1460     PRINT CPOS(5%,6%); DE$; HI$; CP; "V CALIBRATION" ;E0$;
1470     IF CP < 160 THEN PRINT @2%, "R"; I% - 1%; ","
1480     IF CP > 160 THEN PRINT @2%, "R4,"
1490     PRINT @1%, CP;"V,N" \ WAIT 4000
1500     PRINT @2%, "?" \ INPUT @2%, R
1510     PRINT @2%, "?" \ INPUT @2%, R1
```

```
1520      T% = T% + 1% \ IF T% > 5% THEN GOTO 1540
1530      IF R1 < R - (R1 * 5E-6) OR R1 > R + (R1 * 5E-6) THEN GOTO 1500
1540      PRINT @2%, "?" \ INPUT @2%, R$
1550      PRINT @3%, "N"; R$; "P2CO"
1560      WAIT FOR SRQ
1570    NEXT I%
1580 PRINT @1%, "+0V,S"
1590 CF$(2%) = "    DC CALIBRATION COMPLETE" \ Y% = 1%
1600 IF Z% = 1% THEN GOTO 2000 ELSE 30110
2000 REM ************************ AC CAL *******************************
2010 PRINT @3,"F2"
2020 IF AC% = 0% AND Z% = 1% THEN GOTO 3000
2030 IF AC% = 0% AND Z% = 0% THEN GOTO 2420
2040 PRINT AD$; CPOS(2%,7%); "AC CAL";
2050 IF Z% = 1% THEN GOTO 2110
2060 PRINT CPOS(4%,5%); "Connect the output of the 5100B to"
2070 PRINT CPOS(5%,5%); "the input of the 8840A and the 8506A."
2080 PRINT CPOS(8%,33%); EO$;
2090 PRINT CPOS(8%,8%); RI$; "Touch SCREEN to continue"; EO$; BP$;
2100 K% = KEY \ WAIT FOR KEY \ K% = KEY
2110 PRINT CPOS(3%,7%); DE$; BL$; "PERFORMING RANGE CALIBRATION"; EO$
2120 PRINT @2,"VA2RT0,"
2130    FOR I% = 1% TO 6%
2140      PRINT @3%, "G2"
2150      INPUT @3%, CP
2160      PRINT CPOS(5%,6%); LE$; HI$; CP; "V CALIBRATION" ;EO$;
2170      PRINT @1%, CP; "V, 400H, N"
2180      PRINT @2%, "@" \ WAIT FOR SRQ
2190      INPUT @2%, DU$
2200      PRINT @2%, "@" \ WAIT FOR SRQ
2210      INPUT @2%, G$
2220      PRINT @3%, "N"; G$; "P2CO"
2230      WAIT FOR SRQ
2240    NEXT I%
2250 PRINT @3%, "C3C2"
2260 WAIT FOR SRQ
2270 PRINT CPOS(3%,7%); DE$; BL$; "PERFORMING H/F CALIBRATION"; EO$
2280    FOR I% = 1% TO 3%
2290      PRINT @3%, "G2"
2300      INPUT @3%, CP
2310      PRINT CPOS(5%,6%); LE$; HI$; CP; "V CALIBRATION" ;EO$;
2320      PRINT @1%, CP; "V, 50E3H, N"
2330      PRINT @2%, "@" \ WAIT FOR SRQ
2340      INPUT @2%, DU$
2350      PRINT @2%, "@" \ WAIT FOR SRQ
2360      INPUT @2%, G$
2370      PRINT @3%, "N"; G$; "P2CO"
2380      WAIT FOR SRQ
2390    NEXT I%
2400 PRINT @1%, "+0V,S"
2410 CF$(3%) = "    AC CAL COMPLETE"
2420 Y1% = 11%
2430 IF Z% = 1% THEN GOTO 3000 ELSE 30110
3000 REM ********************** OHMS CAL *******************************
3010 DIM Z0$(6%), Z1$(6%), Z0(6%), Z0$(6%)
3020 Z0(1%) = 100 \ Z0(2%) = 1E3 \  Z0(3%) = 10E3 \ Z0(4%)=100E3
3030 Z0(5%)=1E6 \ Z0(6%)=10E6
3040 Z0$(1%) = "100" \ Z0$(2%) = "1k" \ Z0$(3%) = "10 k"  \ Z0$(4%) = "100 k"
3050 Z0$(5%) = "1000 k" \ Z0$(6%) = "10 M"
3060 PRINT AD$; CPOS(2%,7%); "OHMS CAL";
```

```
3070 GOSUB 6570
3080 PRINT CPOS(3%,4%); DE$; "Remove the short from the"
3090 PRINT CPOS(4%,4%); "8506A.  Connect the Sense leads"
3100 PRINT CPOS(5%,4%); "of the 8506A to to Source leads"
3110 PRINT CPOS(6%,4%); "of the 5100B."
3120 PRINT CPOS(8%,33%); EO$;
3130 PRINT CPOS(8%,8%); RI$; "Touch SCREEN to continue"; EO$; BP$;
3140 K% = KEY \ WAIT FOR KEY \ K% = KEY
3150 ! $$$$$$$$$$$$$$$$$$$ 8506A 2 TERM OHMS DATA $$$$$$$$$$$$$$$$$$$$$$$$$$
3160 PRINT CPOS(8%,4%); DE$; BL$; "Reading ACTUAL resistance values"; EO$;
3170   FOR ZZ% = 1% TO 6%
3180     PRINT @2%, "R"; ZZ%; "KNP"; Z2(ZZ%); "P,"
3190     PRINT @1%, Z0(ZZ%); "Z, N" \ WAIT 4000
3200     PRINT @2%, "?" \ INPUT @2%, DU$
3210     PRINT @2%, "?" \ INPUT @2%, Z0$(ZZ%)
3220   NEXT ZZ%
3230 ! $$$$$$$$$$$$$$$$$$$$$ CAL 8840A 2 TERM OHMS $$$$$$$$$$$$$$$$$$$$$$$$$
3240 PRINT CPOS(3%,4%); DE$; "Disconnect from the 8506A and connect"
3250 PRINT CPOS(4%,4%); "to the 8840A using the same cable."
3260 PRINT CPOS(8%,33%); EO$;
3270 PRINT CPOS(8%,8%); RI$; "Touch SCREEN to continue"; EO$; BP$;
3280 K% = KEY \ WAIT FOR KEY \ K% = KEY
3290 PRINT CPOS(3%,4%); DE$; BL$; "Performing 2 terminal Ohms cal"; EO$;
3300    FOR ZZ% = 1% TO 6%
3310      PRINT @3%, "G2"
3320      INPUT @3%, CP
3330      PRINT CPOS(5%,4%); LE$; HI$; Z0$(ZZ%); " OHMS CALIBRATION";
3340      PRINT @1%, CP; "Z, N" \ WAIT 4000
3350      PRINT @3%, "N"; Z0$(ZZ%); "P2C0"
3360      WAIT EOR SRQ
3370    NEXT ZZ%
3380 GOSUB 6860
3390 ! $$$$$$$$$$$$$$$$$ 8506A 4 TERM OHMS DATA $$$$$$$$$$$$$$$$$$$$$$$$$$
3400 PRINT CPOS(3%,4%); DE$; "Remove the short from the"
3410 PRINT CPOS(4%,4%); "8506A.  Connect the Source and"
3420 PRINT CPOS(5%,4%); "Sense leads of the 8506A to"
3430 PRINT CPOS(6%,4%); "the output of the 5100B."
3440 PRINT CPOS(8%,33%); EO$;
3450 PRINT CPOS(8%,8%); RI$; "Touch SCREEN to continue"; EO$; BP$;
3460 K% = KEY \ WAIT FOR KEY \ K% = KEY
3470 PRINT @1%, "100Z, X1,"
3480 PRINT CPOS(8%,4%); DE$; BL$; "Reading ACTUAL Resistance values"; EO$;
3490   FOR ZZ% = 1% TO 6%
3500     PRINT @2%, "R"; ZZ%; "KNP"; Z4(ZZ%); "P,"
3510     PRINT @1%, Z0(ZZ%); "Z, N" \ WAIT 4000
3520     PRINT @2%, "?" \ INPUT @2%, DU$
3530     PRINT @2%, "?" \ INPUT @2%, Z1$(ZZ%)
3540   NEXT ZZ%
3550 ! $$$$$$$$$$$$$$$$$ 8840A 4 TERM OHMS CAL $$$$$$$$$$$$$$$$$$$$$$$$$$$$$
3560 PRINT CPOS(3%,4%); DE$; "Disconnect from the 8506A and connect"
3570 PRINT CPOS(4%,4%); "to the 8840A using the same cables."
3580 PRINT CPOS(8%,33%); EO$;
3590 PRINT CPOS(8%,8%); RI$; "Touch SCREEN to continue"; EO$; BP$;
3600 K% = KEY \ WAIT FOR KEY \ K% = KEY
3610 PRINT @1%, "100Z, X1,"
3620 PRINT CPOS(3%,4%); DE$; BL$; "Peforming 4 terminal OHMS cal"; EO$;
3630    FOR ZZ% = 1% TO 6%
3640      PRINT @3%, "G2"
3650      INPUT @3%, CP
3660      PRINT CPOS(5%,4%); LE$; HI$; Z0$(ZZ%); " OHMS CALIBRATION"; EO$;
```

```
3670       PRINT @1%, CP; "Z, N" \ WAIT 4000
3680       PRINT @3%, "N"; Z1$(ZZ%); "P2C0"
3690       WAIT FOR SRQ
3700    NEXT ZZ%
3710 PRINT @1%, "X0,+0V,S"
3720 CF$(4%) = "    OHMS CAL COMPLETE" \ Y2% = 21%
3730 IF Z% = 1% THEN GOTO 4000 ELSE 30110
4000 REM ******************** DC CURRENT CAL *******************************
4010 PRINT AD$; CPOS(2%,7%); "DC CURRENT CAL"
4020 PRINT CPOS(4%,5%); DE$; "Connect the output of the 5100B"
4030 PRINT CPOS(5%,5%); "to the current input of the 8840A."
4040 PRINT CPOS(8%,33%); E0$;
4050 PRINT CPOS(8%,8%); RI$; "Touch SCREEN to continue"; E0$; BP$;
4060 K% = KEY \ WAIT FOR KEY \ K% = KEY
4070 PRINT CPOS(3%,7%); DE$; BL$; "PERFORMING RANGE CALIBRATION"; E0$;
4080 PRINT @3%, "F5G2"
4090 INPUT @3%, CP
4100 PRINT CPOS(5%,6%); LE$; HI$; CP;"A CALIBRATION" ;E0$;
4110 PRINT @1%, "1Z, X0, N" \ WAIT 1000
4120 PRINT @3%, "C0"
4130 WAIT FOR SRQ
4140 PRINT @3%, "G2"
4150 INPUT @3%, CP
4160 PRINT CPOS(5%,6%); LE$; HI$; CP;"A CALIBRATION" ;E0$;
4170 PRINT @1%, CP;"A, N" \ WAIT 5000
4180 PRINT @3%, "C0"
4190 WAIT FOR SRQ
4200 PRINT @1%, "+0V,S"
4210 CF$(5%) = "    DC CURRENT CAL COMPLETE" \ Y3% = 31%
4220 IF Z% = 1%.THEN GOTO 5000 ELSE 30110
5000 REM ******************** AC CURRENT CAL *******************************
5010 PRINT @3,"F6"
5020 IF AC% = 0% AND Z% = 1% THEN GOTO 5260
5030 IF AC% = 0% AND Z% = 0% THEN GOTO 5260
5040 PRINT AD$; CPOS(2%,7%); "AC CURRENT CAL";
5050 IF Z% = 1% THEN GOTO 5110
5060 PRINT CPOS(4%,5%); DE$; "Connect the output of the 5100B"
5070 PRINT CPOS(5%,5%); "to the current input of the 8840A."
5080 PRINT CPOS(8%,33%); E0$;
5090 PRINT CPOS(8%,8%); RI$; "Touch SCREEN to continue"; E0$;
5100 K% = KEY \ WAIT FOR KEY \ K% = KEY
5110 PRINT CPOS(3%,7%); DE$; BL$; "PERFORMING RANGE CALIBRATION"; E0$;
5120 PRINT @3%, "F6G2"
5130 INPUT @3%, CP
5140 PRINT CPOS(5%,6%); LE$; H1$; ".1A CALIBRATION" ;E0$;
5150 PRINT @1%, CP; "A 1E3H, N" \ WAIT 5000
5160 PRINT @3%, "C0"
5170 WAIT FOR SRQ
5180 PRINT @3%, "G2"
5190 INPUT @3%, CP
5200 PRINT CPOS(5%,6%); LE$; HI$; "1A CALIBRATION" ;E0$;
5210 PRINT @1%, CP;"A 1E3H, N" \ WAIT 5000
5220 PRINT @3%, "C0"
5230 WAIT FOR SRQ
5240 PRINT @1%, "+0V,S"
5250 CF$(6%) = "    AC CURRENT CAL COMPLETE"
5260 Y4% = 41% \ GOTO 30110
5270 PRINT AD$; NS$; CPOS(1%,1%)                ! CLEAR DISPLAY
5280 END
5290 !
```

```
6000 REM ************************ SUBROUTINES ****************************
6010 !
6020 ! &&&&&&&&&&&&&&&&&&&&&&&& DC ZERO &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
6030 ! $$$$$$$$$$$$$ ZERO 8506A AND 8840A FOR A/D CAL $$$$$$$$$$$$$$$$$$$$$$
6040 PRINT CPOS(4%,5%);DE$;"Apply a good quality short to the"
6050 PRINT CPOS(5%,5%);"front panel input terminals of"
6060 PRINT CPOS(6%,5%);"the 8505A and the 8840A."
6070 PRINT CPOS(8%,33%); EO$;
6080 PRINT CPOS(8%,8%); RI$; "Touch SCREEN to continue"; EO$; BP$;
6090 K% = KEY \ WAIT FOR KEY \ K% = KEY
6100 PRINT CPOS(4%,1%); DE$;
6110    IF T% = 1% THEN PRINT CPOS(7%,6%); EO$; LE$; "FIRST ITERATION"
6120    IF T% = 2% THEN PRINT CPOS(7%,6%); EO$; LE$; "SECOND ITERATION"
6130 PRINT @3%, "G2"
6140 INPUT @3%, CP
6150 PRINT CPOS(5%,6%); LE$; HI$; CP; "V CALIBRATION" ;EO$;
6160 PRINT @1%, CP; "V,N" \ WAIT 3000
6170 PRINT CPOS(8%,3%); BL$; "Now zeroing the 8506A and the 8840A"; EO$; DE$;
6180 PRINT @3%, "CO"
6190 PRINT @2%, "VROF20,"
6200   FOR ZE% = 0% TO 1%
6210     PRINT @2%, "R"; ZE%; "?" \ INPUT @2%, DU$
6220     PRINT @2%, "KO?" \ INPUT @2%, DU$
6230   NEXT ZE%
6240 PRINT CPOS(4%,5%); DE$; "Connect the 8840A and the 8506A"
6250 PRINT CPOS(5%,5%); "to the 5100B output terminals."
6260 PRINT CPOS(8%,33%); EO$;
6270 PRINT CPOS(8%,8%); RI$; "Touch SCREEN to continue"; EO$; BP$;
6280 K% = KEY \ WAIT FOR KEY \ K% = KEY
6290 PRINT CPOS(4%,1%); DE$;
6300 RETURN
6310 ! $$$$$$$$$$$$$ ZERO 8506A AND 8840A FOR RANGE CAL $$$$$$$$$$$$$$$$$$$$$$
6320 PRINT CPOS(4%,5%);DE$;"Apply a good quality short to the"
6330 PRINT CPOS(5%,5%);"front panel input terminals of"
6340 PRINT CPOS(6%,5%);"the 8506A and the 8840A."
6350 PRINT CPOS(8%,33%); EO$;
6360 PRINT CPOS(8%,8%); RI$; "Touch SCREEN to continue"; EO$; BP$;
6370 K% = KEY \ WAIT FOR KEY \ K% = KEY
6380 PRINT CPOS(4%,1%); DE$;
6390 PRINT @3%, "G2"
6400 INPUT @3%, CP
6410 PRINT CPOS(5%,6%); LE$; HI$; CP; "V CALIBRATION" ;EO$;
6420 PRINT CPOS(8%,3%); BL$; "Now zeroing the 8506A and the 8840A"; EO$; DE$;
6430 PRINT @1%, CP; "V,N" \ WAIT 3000
6440 PRINT @3%, "CO"
6450 PRINT @2%, "VROF20,"
6460   FOR ZE% = 0% TO 4%
6470     PRINT @2%, "R"; ZE%; "?" \ INPUT @2%, DU$
6480     PRINT @2%, "KO?" \ INPUT @2%, DU$
6490   NEXT ZE%
6500 PRINT CPOS(4%,5%); DE$; "Connect the 8840A and the 8506A"
6510 PRINT CPOS(5%,5%); "to the 5100B output terminals."
6520 PRINT CPOS(8%,33%); EO$;
6530 PRINT CPOS(8%,8%); RI$; "Touch SCREEN to continue"; EO$; BP$;
6540 K% = KEY \ WAIT FOR KEY \ K% = KEY
6550 PRINT CPOS(4%,1%); DE$;
6560 RETURN
6570 ! &&&&&&&&&&&&&&&&&&&&&&&&&&&&& OHMS ZERO &&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
6580 DIM Z2(6%), Z4(6%)
6590 ! $$$$$$$$$$$$$$$$$$$$$$$$$$ 2 TERMINAL ZERO $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
6600 PRINT CPOS(3%,4%); EO$; DE$; "Connect a high quality TWO TERMINAL"
6610 PRINT CPOS(4%,5%); "SHORT to the front panel of the"
6620 PRINT CPOS(5%,5%); "8506A and the 8840A.  Place the"
6630 PRINT CPOS(6%,5%); "8506A in TWO TERMINAL operation."
6640 GOTO 6680
6650 PRINT CPOS(3%,4%); EO$; DE$; "Make sure the 8506A is in TWO"
6660 PRINT CPOS(4%,5%); "TERMINAL ohms operation."; EO$
6670 PRINT CPOS(8%,33%); EO$;
6680 PRINT CPOS(8%,8%); RI$; "Touch SCREEN to continue"; EO$; BP$;
6690 K% = KEY \ WAIT FOR KEY \ K% = KEY
6700 PRINT CPOS(3%,4%); DE$;
6710 PRINT @3%, "F3G2"
6720 INPUT @3%, CP
6730 PRINT CPOS(5%,4%); LE$; HI$; CP; "OHMS CALIBRATION"; EO$
6740 PRINT CPOS(8%,8%); DE$;
6750 PRINT CPOS(8%,2%); BL$; "Now ZEROING the 8506A and the 8840A"; EO$;
6760 PRINT @2%, "ZR1F20,?" \ INPUT @2%, DU$
6770 PRINT @2%, "G1?" \ INPUT @2%, ST
6780 IF ST>= 27000 THEN GOTO 6650
6790 PRINT @3%, "C0"
6800 !   WAIT FOR SRQ
6810    FOR ZE% = 1% TO 6%
6820      PRINT @2%, "R"; ZE%; "?" \ INPUT @2%, DU$
6830      PRINT @2%, "?" \ INPUT @2%, Z2(ZE%)
6840    NEXT ZE%
6850 RETURN
6860 ! $$$$$$$$$$$$$$$$$$$$$$ 4 TERMINAL ZERO $$$$$$$$$$$$$$$$$$$$$$$$$$
6870 PRINT CPOS(3%,3%); EO$; DE$; "Connect a high quality FOUR TERMINAL"
6880 PRINT CPOS(4%,4%); "SHORT to the front panel of the"
6890 PRINT CPOS(5%,4%); "8506A and the 8840A.  Place the"
6900 PRINT CPOS(6%,4%); "8506A in FOUR TERMINAL operation."
6910 PRINT CPOS(8%,33%); EO$;
6920 PRINT CPOS(8%,8%); RI$; "Touch SCREEN to continue"; EO$; BP$;
6930 K% = KEY \ WAIT FOR KEY \ K% = KEY
6940 PRINT CPOS(3%,4%); DE$;
6950 PRINT @3%, "F4G2"
6960 INPUT @3%, CP
6970 PRINT CPOS(5%,4%); LE$; HI$; CP; "OHMS CALIBRATION"; EO$
6980 PRINT CPOS(8%,8%); DE$;
6990 PRINT CPOS(8%,2%); BL$; "Now ZEROING the 8506A and the 8840A"; EO$;
7000 PRINT @2%, "R1P0?" \ INPUT @2%, DU$
7010 PRINT @3%, "C0"
7020   !  WAIT FOR SRQ
7030    FOR ZE% = 1% TO 6%
7040      PRINT @2%, "R"; ZE%; "?" \ INPUT @2%, DU$
7050      PRINT @2%, "?" \ INPUT @2%, Z4(ZE%)
7060    NEXT ZE%
7070 RETURN
8000 REM ********************* ERROR SRQ ROUTINE ************************
8010 IF SPL(2%)=64% THEN 8030
8020 IF SPL(3%) = 72% THEN GOTO 8030 ELSE 8040
8030 RESUME
8040 PRINT AD$; CPOS(1%,1%); BP$
8050 PRINT "8840A ERROR OCCURED" \ INPUT @3%, ER$
8060 PRINT @3,"G7" \ INPUT @3, ER
8070 ER = ER - 1000
8080 IF AC% = 0% THEN CF$(3%) = "AC CONVERTER NOT INSTALLED"
8090 IF AC% = 0% THEN CF$(6%) = "AC CONVERTER NOT INSTALLED"
8100 IF ER = 30 THEN 8030
8110 PRINT "ERROR NUMBER ";ER
```

```
8120 RESUME 5280
9000 REM ******************** SELF TEST ROUTINE *************************
9010 TE% = 0%
9020 PRINT CPOS(1%,1%); DS$;
9030 PRINT CPOS(1%,34%); EO$;
9040 PRINT CPOS(1%,1%); EO$; UL$; "Performing internal diagnostics"; EO$
9050 PRINT
9060 PRINT @3%, "N48P1Z0"
9070 Y1% = SPL(3%)
9080 WAIT FOR SRQ
9090 Y% = SPL(3%)
9100 IF Y% = 80% THEN GOTO 9160
9110 TE% = 1%
9120 PRINT @3%, "G7" \ INPUT @3%, ER
9130 ER = ER - 1000
9140 PRINT "Failed test number ";ER
9150 GOTO 9080
9160 Y1% = SPL(3%)
9170 INPUT @3%, DU
9180 IF DU >= 1E21 THEN GOTO 9070
9190 PRINT @3%, "NOP1"
9200 IF TE% = 1% GOTO 5280
9210 PRINT CPOS(4%,17%); HI$; "PASSED"; EO$; BP$; \ WAIT 5000
9220 RETURN
30000 REM ****************** SELECT CAL FUNCTION *********************
30010 PRINT @3%,"G4" \ INPUT @3%, CS
30020 IF CS = 1000 THEN GOTO 540
30030 PRINT AD$; DS$                        ! CLEAR SCREEN, DOUBLE SIZE
30040 IF Y%+Y1%+Y2%+Y3%+Y4% = 105% THEN GOTO 30110
30050 PRINT CPOS(2%,6%); CF$(1%)
30060 PRINT CPOS(7%,11%);EO$; CPOS(7%,5%); RI$; "YES";EO$;
30070 PRINT CPOS(7%,36%); EO$; CPOS(7%,32%); RI$; "NO "; EO$;
30080 K% = KEY \ WAIT FOR KEY \ K% = KEY
30090 IF K%<>51% AND K%<>60% GOTO 30080
30100 IF K% = 51% THEN Z% = 1% \ GOTO 1000
30110 PRINT EO$; AD$
30120    FOR I% = 2% TO 7%
30130       PRINT CPOS(I%,6%); CF$(I%);
30140    NEXT I%
30150 IF Y%+Y1%+Y2%+Y3%+Y4% = 105% THEN GOTO 30370
30160 K% = KEY \ WAIT FOR KEY \ K% = KEY
30170 X% = 2%
30180    FOR I% = 1% TO 61% STEP 10%
30190       IF K% = I% THEN GOTO 30230
30200       X% = X% + 1%
30210    NEXT I%
30220 GOTO 30160
30230 IF K% = Y% OR K% = Y1% OR K% = Y2% OR K% = Y3% OR K% = Y4% GOTO 30160
30240 PRINT CPOS(X%,7%); "X";
30250 IF K% = 51% GOTO 30380
30260 PRINT CPOS(7%,1%); DE$; CPOS(7%,15%); EO$; CPOS(7%,8%); RI$; CF$(8%); EO$
30270 PRINT CPOS(7%,36%); EO$; CPOS(7%,29%); RI$; CF$(9%);EO$
30280 K% = KEY \ WAIT FOR KEY \ K% = KEY
30290 IF K% = 52% OR K% = 53% GOTO 30320
30300 IF K% = 59% OR K% = 60% GOTO 30110
30310 GOTO 30280
30320 IF X% = 2% THEN GOTO 1000
30330 IF X% = 3% THEN GOTO 2000
30340 IF X% = 4% THEN GOTO 3000
30350 IF X% = 5% THEN GOTO 4000
```

```
30360 IF X% = 6% THEN GOTO 5000
30370 PRINT CPOS(7%,1%); DE$  \ WAIT 5000
30380 GOTO 5270
32762 REM ********************* SAVE PROGRAM ***************************
32763 KILL"MFO:40CAL2.BAL"\SAVEL"MFO:40CAL2"
32764 KILL"MFO:40CAL2.BAS"\SAVE"MFO:40CAL2"
32765 KILL"EDO:40CAL2.BAL"\SAVEL"EDO:40CAL2"
32766 KILL"EDO:40CAL2.BAS"\SAVE"EDO:40CAL2"
32767 END
```

SOFTWARE: SAMPLE APPLICATIONS PROGRAMS

THE FOLLOWING PROGRAM IS PRINTED WITH PERMISSION OF THE HEWLETT PACKARD COMPANY, LAKE STEVENS INSTRUMENT DIVISION. THIS PIECE OF SOFTWARE IS WRITTEN IN A STRUCTURED FORMAT AND IS DESIGNED TO WORK UNDER A PLANT WIDE EXECUTIVE PROGRAM, WHICH ALLOWS MULTIPLE ACCESS TO COMPILED DATA.

THE IMPORTANT CONCEPTS TO BE STUDIED HERE IS THE STRUCTURED LAYOUT, USE OF REMARKS, AND USER INTERFACE --- NOT THE CODE DETAILS.

THIS IS AN EXAMPLE OF A WELL WRITTEN, SHORT ATE PROGRAM.

NOTES:

```
1  ! LAST MOD 18:47:38   9 May 1986 MURPH
2  !  RE-STORE "/INSTRUMENT/3325A/TEST75"
3  !  RE-STORE "/USERS/DANIELSLEE/MURPHY/3325_TESTS/TEST75_MAY09"
4!    OUTPUT KBD USING "#,K"; "REN 1,1 XINDENT X S D X"
5     COM /User_common1/ U(1:50),U$[300]
6     COM /Hv_mode/ Hv$[10]
7     DIM Misc(2)
8     Misc(1)=0
9     DIM Ydata(2)
10    DIM Xdata(2)
11    DIM X$[2]
12    DIM Data$[2]
13    FOR I=1 TO 34
14      Datapt=I
15      CALL Test75(Datapt,Datum,X,Datum$,X$,Misc(*),Flag,Ydata(*),Xdata(*),Data
$,F1,F2,F3,F4,R)
16      PRINT Datapt,Datum,Datum$
17      WAIT .5
18    NEXT I
19    END
20    !
21    !
```

```
22 Test75:SUB Test75(Datapt,Datum,X,Datum$,X$,Misc(*),Flag,Ydata(*),Xdata(*),Dat
a$,F1,F2,F3,F4,R)
23    !##########################################################################
24    !             TEST75    3325A SCOPE MEASUREMENTS
25    !##########################################################################
26    !**************************************************************************
27    !*************************** INDEX ****************************************
28       !
29 Index:!  The following are labels used to find subs or particular areas
30       !  within this test.  To use just delete the line # and the ! and
31       !  Execute the line.  To get back to this point just type "EDIT Index".
32       !
33!     EDIT Test75
34                               ! SUB  Called by Executive for each Datapt.
35!     EDIT Modify_param
36                               ! LABEL For changeing test arguments, suc as Amp
37                               !       Freq, attenation, ect.
38!     EDIT Datapt_select
39                               ! SUB  Fills Datum with readings from Common.
40!     EDIT Assign_devices
41                               ! SUB
42!     EDIT Setup_54110
43                               ! SUB
44!     EDIT Setup_3325
45                               ! SUB
46!     EDIT Measure
47                               ! SUB  makes Undershoot and Flatness measurements
48!     EDIT Cleanup
49                               ! SUB  Last sub called before going back to Exec.
50!     EDIT Format
51                               ! SUB  Formats all measurement results as desired
52!     EDIT Tweak_dut
53                               ! SUB  For making adjustments to inst. before
54                               !      testing.
55!     EDIT Input_kbd
56                               ! SUB  Returns if space bar or enter key are
57                               !      pressed.
58!     EDIT Amp_flatness
59                               ! SUB
60!     EDIT Scope_problem
61                               ! SUB  Operator instruction if HP-IB hang.
62!     EDIT Hv_instruct_1
63                               ! SUB  Operator inst. for HV testing.
64!     EDIT Hv_instruct_2
65                               ! SUB  Operator instruction.
66!     EDIT Connect_inst
67                               ! SUB  Operator instructions for 1st time connect
68!     EDIT Std_inst
69                               ! SUB  Operator inst. for non option instrument.
70!     EDIT Hv_present_inst
71                               ! SUB  Informs user that instrument is a HV inst.
72                               !
73   !***************************** END OF INDEX ********************************
74   !****************************************************************************
75    !
```

```
!##########################################################################
! MINI SPECIFICATIONS
!***********************************
! MODULE OVERVIEW
!*         This program will measure the Frequency, Period, Positive,
!*         Width, Negative Width, Risetime, Falltime, Topbase, Vpp,
!*         Risetime Overshoot, Falltime Overshoot, Dutycycle, Vrms, Vmin,
!*         Vtop, Vbase, Risetime undershoot, and Falltime undershoot.
!*         It will also measure Amplitude flatness of the mode (most
!*         common level(Topbase)), with respect to a reference frequency.
!*         This program is set up to distinguish the difference
!*         between a HV option 3325 and a standard RF.  If
!*         it is an option instrument, common will be changed in order
!*         for the Skip_test and Skip_datapt to be notified (com/hv_mode/
!*         hv$).
!*
!*         This Sub only uses the following Pass parameters, even though
!*         more are passed to allow compatability with Executive:
!*         Datapt, Datum, Datum$, and Misc(1).
!* PROGRAMERS:  Murphy Moore
!*
!* DATE CREATED: 4-08-86
!*************************************************************************
!********************** DATA DICTIONARY **********************************
!INPUTS
!**************
! Datapt        - Data Point should be an integer 1-45 indicating what
!                 measurement is to be passed in the Variable Datum.
! F1-F4         - Not used.
! Flag          - Not used.
! Misc(1)       - Passed by Executive as an integer value of 1 or 0
!                 1 if the first time test was called by Executive, 0 if
!                 test had previously been called. It is then changed in
!                 this sub, indicating that the test has been called.
! R             - Not used.
! X             - Not used.
! Xdata         - Not used.
! Ydata         - Not used.
!***************
! OUTPUTS
!***************
! Data$         - Not used.
! Datum         - The measurement that will be passed back for the
!                 indicated Data Point.
! Datum$        - Units used for Datum (HZ, VPP etc.), these units are
!                 consistant with the value of Datum.
! X$            - Not used.
!
!*************** END OF MINI SPECS AND DATA DICTIONARY *******************
!*************************************************************************
!
```

```
!*************************************************************************
!**************** INTERNAL REFERENCE SPECIFICATIONS **********************
!
! MODULE DISCRIPTION
! This module is set up so that a programer may choose the appropriate
! set up and mesurement parameters for all 3325A scope mesurements.  These
! parameters may be changed by going to Modify_param and change the values
! These values are then passed on to the appropriate subs in order to
! achieve the desired results.
!*************************************************************************
!******************** INTERNAL DATA DICTIONARY ***************************
!
! This Sub makes calls to the following Subs.
!***************
! Assign_devices
! Setup_54110
! Tweak_dut
! Input_kbd
! Measure
! Cleanup
! Datapt_select
!***************
! This Sub program uses the following Variables.
!***************
! Atten          - Attenuation factor for scope probes 1.0 - 1000.0
! Average        - Number of averages made by scope for each reading 1-2048
! Backgrnd_color- Color of scope display background.
! Grid_color     - Integer value for color of grid for scope.
! Hv             - Integer 1 or 0 for whether the instrument is a High
!                  votage instrument or not.
! Hv_flat_amp    - Amplitude in VPP for all high voltage flatness
!                  measurements.
! Hv_flat_freqs - Array of frequency's in HZ for comparison with a
!                  reference frequency.
! Hv_ref_freq    - High voltage reference frequency, used for comparison.
!                  Frequency in HZ.
! Hv_und_freq    - Frequency in HZ used for all measurements, except
! Std_flat_amp   - Amplitude in VPP for all Standard flatness measurements.
! Std_flat_freqs- Array of Frequency's in HZ to be compared to a reference
!                  frequency's amplitude for all Standard flatness
!                  measurements.
! Std_ref_freq   - Frequency in HZ of the Reference frequency, which all
!                  Standard flatness frequency amplitudes will be compared
!                  to.
! Std_und_amp    - Amplitude in VPP for all measurement, other than
!                  flatness, during High voltage measurements.
! Hv_und_amp     - Voltage in VPP for all measurements, except flatness
!                  during High voltage measurements.
! Key_pressed    - Integer 1 or 0 used to determine which key was pressed.
!                  Flatness for Standard measurements.
! Std_und_freq   - Frequency in HZ of all measurements, other than
!                  Flatness for Standard measurements.
! Wave_color     - Color of trace on scope.
!
!************** END OF INTERNAL REFERENCE SPECIFICATIONS *****************
!*************************************************************************
!
```

```
184     !********************************************************************
185     !****************  HIERARCHY CHART FOR 3325A SCOPE TEST  **************
186     !********************************************************************
187     !                          ___________
188     !                         | EXECUTIVE |
189     !                         |___________|
190     !                               | SUB
191     !                               | 1
192     !                            ___|____
193     !                           | TEST75 |
194     !                           |________|
195     !  _____________________________|
196     !  |
197     !  |  SUB                    SUB   ________________
198     !  |   2    ______________    9   | ASSIGN_HPSCOPE |
199     !  |_______|ASSIGN_DEVICES|_______| ASSIGN_IO_DUT  |
200     !  |       |______________|       |________________|
201     !  |
202     !  |                               ________________
203     !  |   3    ______________    _10_| PROB_INST_54110|
204     !  |_______| SETUP_54110  |___|   |________________|
205     !  |       |______________|   |11_|  INPUT_KBD     |
206     !  |                              |________________|
207     !  |
208     !  |                               ________________
209     !  |                          _12_| HV_PRESENT_3325|
210     !  |                          |   |________________|
211     !  |                          |13_|  SETUP_3325    |
212     !  |                          |   |________________|
213     !  |   4    ______________    |14_|  MANUAL_54110  |
214     !  |_______|  TWEAK_DUT   |___|   |________________|
215     !  |       |______________|   |15_| HV_PRESENT_INST|
216     !  |                          |   |________________|
217     !  |                          |16_|  STD_INST      |
218     !  |                              |________________|
219     !  |                          |17_|  CONNECT_INST  |
220     !  |   5    ______________        |________________|
221     !  |_______|   INPUT_KBD  |
222     !  |       |______________|         _______________
223     !  |                           _18_| CHECK_DUT     |
224     !  |                          |    |_______________|
225     !  |                          |_19_| HV_INSTRUCT_1 |
226     !  |                          |    |_______________|
227     !  |                          |_20_|  INPUT_KBD    |
228     !  |                          |    |_______________|
229     !  |                          |_21_|  SETUP_3325   |
230     !  |                          |    |_______________|
231     !  |   6    ______________    |_22_| AUTOSCLE_54110|
232     !  |_______|  MEASURE     |___|    |_______________|
233     !  |       |______________|   |_23_| HV_INSTRUCT_2 |
234     !  |                          |    |_______________|
235     !  |                          |_24_|   ALL_54110   |
236     !  |                          |    |_______________|
237     !  |                          |_25_| UNDERSHT_54110|
238     !  |                          |    |_______________|
239     !  |                          |_26_| AMP_FLATNESS  |
240     !  |                          |    |_______________|
241     !  |                          |_27_|   FORMAT      |
242     !  |                               |_______________|
!  |   7        _______________          __________________
!  |________|   CLEANUP       |___28___|   PRESET_3325    |
!  |        |_________________|        |__________________|
!  |
!  |   8        _______________
!  |________| DATAPT_SELECT|
!           |______________|
!
!
```

```
252   !******************************************************************
253   !***************      PASS PARAMETERS LIST    ***********************
254   !
255   !     SUB              |        INPUT          |        OUTPUT
256   !  __________________  |______________________ |_________________________
257   !        1             | DATAPT,MISC(1)        | DATAPT,DATUM$,MISC(1)
258   !  __________________  |______________________ |_________________________
259   !        2             |    NONE               |        NONE
260   !  __________________  |______________________ |_________________________
261   !        3             | GRID COLOR,           |        NONE
262   !                      | BACKGROUND COLOR,     |
263   !                      | WAVE COLOR,           |
264   !                      | ATTENUATION,          |
265   !                      | NO. OF AVERAGES,COM   |
266   !                      | /HP_SCOPE/@THE54110   |
267   !  __________________  |______________________ |_________________________
268   !        4             | FREQUENCY,AMPLITUDE   | HV_OPTION
269   !                      |COM/DUT_3325_IOPATH/   | COM/HV_MODE/HV$
270   !                      | @THE_3325             |
271   !  __________________  |______________________ |_________________________
272   !        5             |    NONE               |  KEY PRESSED 1 OR 0
273   !  __________________  |______________________ |_________________________
274   !        6             | STD_UND_FREQ,         | USER COMMON1/READINGS_1(45)
275   !                      | STD_UND_AMP,          |
276   !                      | HV_UND_FREQ,          |
277   !                      | HV_UND_AMP,           |
278   !                      | STD_REF_FREQ,         |
279   !                      | STD_FLAT_AMP,         |
280   !                      | STD_FLAT_FREQS(*),    |
281   !                      | HV_REF_FREQ,          |
282   !                      | HV_FLAT_FREQS(*),HV   |
283   !  __________________  |______________________ |_________________________
284   !        7             |COM/DUT_3325_IOPATH/   |        NONE
285   !                      |  @THE_3325
286   !  __________________  |______________________ |_________________________
287   !        8             | DATAPT, COM/USER_     | DATUM,DATUM$
288   !                      | COMMON1/READINGS(*)   |
289   !  __________________  |______________________ |_________________________
290   !        9             |      NONE             | IO/PATH HPSCOPE, DUT
291   !  __________________  |______________________ |_________________________
292   !      10,11           |      NONE             |        NONE
293   !  __________________  |______________________ |_________________________
294   !       12             | @THE_3325             | HV_OR_FR$
295   !  __________________  |______________________ |_________________________
296   !       13             | FREQUENCY,AMPLITUDE   |        NONE
297   !                      | HV_ONOFF$             |
298   !  __________________  |______________________ |_________________________
299   !       14             | FREQUENCY,AMPLITUDE   |        NONE
300   !                      | CHANNEL
301   !  __________________  |______________________ |_________________________
302   !      15-19           |      NONE             |        NONE
303   !  __________________  |______________________ |_________________________
304   !       20             |      NONE             |     KEY_PRESSED
305   !  __________________  |______________________ |_________________________
306   !       21             | FREQUENCY,AMPLITUDE   |        NONE
307   !                      | HV_ONOFF$             |
308   !  __________________  |______________________ |_________________________
309   !      22-23           |      NONE             |        NONE
310   !  __________________  |______________________ |_________________________
```

```
!      24            |      NONE            |    READINGS(16)
!                    |                      |
!  __________________  ____________________  ________________________
!      25            | PERIOD,TOP,BASE      |  U_READINGS(2)
!                    |                      |
!  __________________  ____________________  ________________________
!      26            | FLATNESS_REF,        | F_READINGS(*)
!                    | FREQUENCYS(*)        |
!                    |                      |
!  __________________  ____________________  ________________________
!      27            | READINGS(24)         |  READINGS(24)
!                    | NO_OF_READINGS       |
!                    |                      |
!  __________________  ____________________  ________________________
!      28            | @THE_3325            |       NONE
!                    |                      |
!  __________________  ____________________  ________________________
!
! The pass parameters for the other subs may be found in the subs
! themselves.
!
!********************************************** Murphy M. 4-08-86 ****
!*******************************************************************
!
```

```
330   !*********************************************************************
331 Modify_param:              ! Label for finding Modification section.
332   !
333   ! You may change the value of the folowing variables in order to change
334   ! the the testing conditions of the instrument.
335   ! Remember:  The spec values must be changed in Specfile, Namefile
336   ! and ITD_Subs.
337   !
338   !
339             ! Below are the settings for Undershoot measurements.
340                            !
341     Std_und_freq=1.1*10^6  ! Values in HZ
342     Std_und_amp=10         ! Value in VPP
343     Hv_und_freq=5.5*10^5   ! Value in HZ
344     Hv_und_amp=30          ! Value in VPP
345                            !
346             ! Below are the settings for Flatness measurements.
347                            !
348     Std_ref_freq=1*10^3    ! Value in VPP (Reference frequency)
349     Std_flat_amp=10        ! Value in VPP
350     DIM Std_flat_freqs(1:7)
351     Std_flat_freqs(1)=2*10^5
352     Std_flat_freqs(2)=5*10^5
353     Std_flat_freqs(3)=1*10^6
354     Std_flat_freqs(4)=2*10^6
355     Std_flat_freqs(5)=5*10^6
356     Std_flat_freqs(6)=1*10^7
357                            ! Flatness frequency's, not High voltage.
358                            !
359     Hv_ref_freq=1*10^3     ! Value in VPP (Reference frequency)
360     Hv_flat_amp=40         ! Value in VPP
361     DIM Hv_flat_freqs(1:4)
362     Hv_flat_freqs(1)=2*10^5
363     Hv_flat_freqs(2)=5*10^5
364     Hv_flat_freqs(3)=1*10^6
365                            ! Flatness frequency's, High voltage.
366                            !
367     Atten=2                ! This sets the Attenuation factor.
368                            !
369     Average=8              ! This sets the number of averages to be taken.
370                            !
371   !
372   ! ***************************************************************
373   ! This routine will guarantee that the test will be run from the
374   ! beginning even if only one data point is selected.  This is
375   ! a bit inflexible but the run time for the 54110 is so short
376   ! and all but the F_undershoot, and R_undershoot measurements are
377   ! all made at once so there is not a lot to gain by separating
378   ! the measurements.  A TESTnn could always be made, if a different
379   ! measurement is desired.
380   !
381     IF Misc(1)=0 THEN
382       CALL Assign_devices
383       CALL Setup_54110(Grid_color,Backgrnd_color,Wave_color,Atten,Average)
384 Tweak_routine:       !
385       DISABLE INTR 7
386             ! Disables interupt, so that the instrument may be plugged
387             ! in and adjusted for testing.
388             !
```

```
389        CALL Tweak_dut(Std_und_freq,Std_und_amp,Hv)
390                ! This sets up Dut for Tweaking, and so that the operator
391                ! may visually see the wave.  It also checks to see if the
392                ! 3325 is a HV option.
393                !
394        CALL Input_kbd(Key_pressed)
395                ! This sub will return either a 1 (Space Bar) or 0
396                ! (Return) keys are pressed.
397        IF Key_pressed=0 THEN
398          GOTO Tweak_routine
399        END IF
400                ! Checks to see if operator is ready to go on.
401                !
402        ENABLE INTR 7
403                ! Enables interupts for problems on bus.
404                !
405        CALL Measure(Std_und_freq,Std_und_amp,Hv_und_freq,Hv_und_amp,Std_ref_f
req,Std_flat_amp,Std_flat_freqs(*),Hv_ref_freq,Hv_flat_amp,Hv_flat_freqs(*),Hv)
406                !
407                !
408        CALL Cleanup
409        Misc(1)=1
410      END IF
411      !
412      CALL Datapt_select(Datapt,Datum,Datum$)
413      !
414    SUBEND                            ! END OF SUB TEST56
415      !
416      !
```

```
417 Datapt_select:SUB Datapt_select(Datapt,Datum,Datum$)
418   !#########################################################################
419   ! DATAPOINT SELECT  SUB
420   !#########################################################################
421   ! MINI SPECIFICATIONS
422   !**********************************
423   ! MODULE OVERVIEW
424   ! This must be passed a Data Point by means of variable "Datapt" in the
425   ! form a of an integer from 1 to 45.  A measurement will be returned
426   ! by means of variable "Datum" and a string indicating units, by means
427   ! of variable "Datum$".
428   !
429   ! PROGRAMER: Murphy
430   ! LAST UPDATE: 4-17-86  Murph
431   !*************************************************************************
432   !********************** DATA DICTIONARY **********************************
433   ! INPUTS
434   !***************
435   ! Datapt        - Data Point should be an integer 1-45, a measurement
436   !                corresponding to the Data Point will be returned in
437   !                variable "Datum".
438   !***************
439   ! OUTPUT
440   !***************
441   ! Datum         - Measurement that corresponds to the "Datapt" passed.
442   ! Datum$        - String indicating units in variable "Datum" (VPP etc.).
443   !
444   !****************** END OF MINI SPECS AND DATA DICTIONARY ****************
445   !*************************************************************************
446   !
```

```
447 !*************************************************************************
448 !**************** INTERNAL REFERENCE SPECIFICATIONS **********************
449 ! MODULE DISCRIPTION
450 ! This module uses Case Select Statements in order to establish the
451 ! value of "Datapt", and then takes the measurements stored in Com/
452 ! Users_common1/Readings_1" and puts the particular reading into
453 ! variable "Datum", along with the Units for the particular measurement,
454 ! which is passed back as the variable "Datum$".
455 !*************************************************************************
456 !******************** INTERNAL DATA DICTIONARY ***************************
457 !
458 ! This Sub makes NO CALLS to any other Subs.
459 !
460 ! This Sub uses the following  Variables.
461 !***************
462 ! Readings_1    - An array of measurements that are stored in Common
463 !                 under the name "User_common1".
464 ! U$            - This is also passed in through "User_common1" but is
465 !                 NOT USED.
466 !
467 !************ END OF INTERNAL REFERENCE SPECIFICATIONS *******************
468 !*************************************************************************
469 !
470   COM /User_common1/ Readings_1(*),U$
471 !
472 !
473 !*************************************************************************
474 ! A CASE will be selected dependent upon the value of the Datapt.
475 !
476 !
477   SELECT Datapt
478   CASE 1                    ! Frequency
479     Datum=Readings_1(1)
480     Datum$="KHZ"
481   CASE 2                    ! Period
482     Datum=Readings_1(2)
483     Datum$="nSec"
484   CASE 3                    ! Positive Width
485     Datum=Readings_1(3)
486     Datum$="nSec"
487   CASE 4                    ! Negative Width
488     Datum=Readings_1(4)
489     Datum$="nSec"
490   CASE 5                    ! Risetime
491     Datum=Readings_1(5)
492     Datum$="nSec"
493   CASE 6                    ! Falltime
494     Datum=Readings_1(6)
495     Datum$="nSec"
496   CASE 7                    ! TopBase
497     Datum=Readings_1(7)
498     Datum$="V"
499   CASE 8                    ! Vpp
500     Datum=Readings_1(8)
501     Datum$="Vpp"
502   CASE 9                    ! Falltime Overshoot
503     Datum=Readings_1(9)
504     Datum$="%"
505   CASE 10                   ! Risetime Overshoot
```

```
506         Datum=Readings_1(10)
507         Datum$="%"
508       CASE 11                   ! Dutycycle
509         Datum=Readings_1(11)
510         Datum$="%"
511       CASE 12                   ! VRMS
512         Datum=Readings_1(12)
513         Datum$="VRMS"
514       CASE 13
515         Datum=Readings_1(13)    ! Vmax
516         Datum$="V"
517       CASE 14                   ! Vmin
518         Datum=Readings_1(14)
519         Datum$="V"
520       CASE 15                   ! VTop
521         Datum=Readings_1(15)
522         Datum$="V"
523       CASE 16                   ! VBase
524         Datum=Readings_1(16)
525         Datum$="V"
526       CASE 17                   ! Risetime Undershoot
527         Datum=Readings_1(17)
528         Datum$="%"
529       CASE 18                   ! Falltime Undershoot
530         Datum=Readings_1(18)
531         Datum$="%"
532       CASE 19                   ! Flatness 200KHZ 10VPP
533         Datum=Readings_1(19)
534         Datum$="%"
535       CASE 20                   ! Flatness 500KHZ 10VPP
536         Datum=Readings_1(20)
537         Datum$="%"
538       CASE 21                   ! Flatness 1MHZ 10VPP
539         Datum=Readings_1(21)
540         Datum$="%"
541       CASE 22                   ! Flatness 2MHZ 10VPP
542         Datum=Readings_1(22)
543         Datum$="%"
544       CASE 23                   ! Flatness 5MHZ 10VPP
545         Datum=Readings_1(23)
546         Datum$="%"
547       CASE 24                   ! Flatness 10MHZ 10VPP
548         Datum=Readings_1(24)
549         Datum$="%"
550       CASE 25                   ! Frequency
551         Datum=Readings_1(25)
552         Datum$="KHZ"
553       CASE 26                   ! Period
554         Datum=Readings_1(26)
555         Datum$="nSec"
556       CASE 27                   ! Positive Width
557         Datum=Readings_1(27)
558         Datum$="nSec"
559       CASE 28                   ! Negative Width
560         Datum=Readings_1(28)
561         Datum$="nSec"
562       CASE 29                   ! Risetime
563         Datum=Readings_1(29)
564         Datum$="nSec"
565       CASE 30                   ! Falltime
```

```
566        Datum=Readings_1(30)
567        Datum$="nSec"
568      CASE 31                      ! TopBase
569        Datum=Readings_1(31)
570        Datum$="V"
571      CASE 32                      ! Vpp
572        Datum=Readings_1(32)
573        Datum$="Vpp"
574      CASE 33                      ! Falltime Overshoot
575        Datum=Readings_1(33)
576        Datum$="%"
577      CASE 34                      ! Risetime Overshoot
578        Datum=Readings_1(34)
579        Datum$="%"
580      CASE 35                      ! Dutycycle
581        Datum=Readings_1(35)
582        Datum$="%"
583      CASE 36                      ! VRMS
584        Datum=Readings_1(36)
585        Datum$="VRMS"
586      CASE 37
587        Datum=Readings_1(37)       ! Vmax
588        Datum$="V"
589      CASE 38                      ! Vmin
590        Datum=Readings_1(38)
591        Datum$="V"
592      CASE 39                      ! VTop
593        Datum=Readings_1(39)
594        Datum$="V"
595      CASE 40                      ! VBase
596        Datum=Readings_1(40)
597        Datum$="V"
598      CASE 41                      ! Risetime Undershoot
599        Datum=Readings_1(41)
600        Datum$="%"
601      CASE 42                      ! Falltime Undershoot
602        Datum=Readings_1(42)
603        Datum$="%"
604      CASE 43                      ! Flatness 200KHZ 40VPP
605        Datum=Readings_1(43)
606        Datum$="%"
607      CASE 44                      ! Flatness 500KHZ 40VPP
608        Datum=Readings_1(44)
609        Datum$="%"
610      CASE 45                      ! Flatness 1MHZ 40VPP
611        Datum=Readings_1(45)
612        Datum$="%"
613        !
614      END SELECT
615        !
616        !
617    SUBEND                               ! END OF SUB DATAPT_SELECT
618      !
619      !
```

```
620 Assign_devices:SUB Assign_devices
621   !#########################################################################
622   ! ASSIGN DEVICES SUB PROGRAM
623   !#########################################################################
624   ! MINI SPECIFICATIONS
625   !******************************
626   ! MODULE OVERVIEW
627   ! This subprogram will assign I/O paths to all of the devices involved
628   ! in this test. It will also test and intialize the DUT in order to
629   ! determine if it is ready to test.
630   !*************************************************************************
631   !*************************** DATA DICTIONARY *****************************
632   ! INPUTS
633   !***************
634   ! NONE
635   !***************
636   ! OUTPUTS
637   !***************
638   ! NONE
639   !
640   !************* END OF MINI SPECS AND DATA DICTIONARY *********************
641   !*************************************************************************
642   !
```

```
643   !************************************************************************
644   !************* INTERNAL REFERENCE SPECIFICATIONS ************************
645   ! MODULE DISCRIPTION
646   ! This Module calls the Subs that assign the desired I/O Paths for the
647   ! equipment being used.
648   !************************************************************************
649   !***************** INTERNAL DATA DICTIONARY *****************************
650   !
651   ! This Sub makes calls to the following Subs.
652   !***************
653   ! Assign_hpscope
654   ! Assign_io_dut
655   !***************
656   ! This Sub uses NO VARIABLES.
657   !
658   !**************** END OF INTERNAL DATA DICTIONARY ***********************
659   !************************************************************************
660   !
661   ! PRINT "Stub for Assign_devices"
662     CALL Assign_hpscope
663     CALL Assign_io_dut
664   SUBEND
665       !
666       !
```

```
667 Setup_54110:SUB Setup_54110(Grid_color,Backrnd_color,Wave_color,Atten,Averag
e)
668     !#########################################################################
669     ! SETUP  54110  SUBPROGRAM
670     !#########################################################################
671     ! MINI SPECIFICATIONS
672     !****************************
673     ! MODULE OVERVIEW
674     ! This sub resets the 54110, and sets up the desired settings for
675     ! measurement. This Sub must be passed the Attenuation 1.0-1000.0 (Atten),
676     ! and number of averages to be made by scope 1-2048 (Averages).
677     !
678     ! PROGRAMMER: Murphy
679     ! LAST MOD: 4-17-86   MM
680     !*************************************************************************
681     !********************** DATA DICTIONARY **********************************
682     ! INPUTS
683     !***************
684     !
685     ! Atten          - Attenuation factor for the scope probes 1.0 - 1000.0.
686     ! Average        - Number of averages made by scope for each reading 1-2048
687     ! Backgrnd_color- Color of scope display background.
688     ! Grid_color     - Color of scope grid.
689     ! Wave_color     - Color of trace on scope.
690     !***************
691     !OUTPUTS
692     !***************
693     ! NONE
694     !
695     !**************** END OF MINI SPECS AND DATA DICTIONARY *****************
696     !*************************************************************************
697     !
```

```
!************************************************************************
!**************** INTERNAL REFERENCE SPECIFICATIONS **********************
!
! MODULE DISCRIPTION
! This module  Resets the  Scope and sets it up to the disired format
! dependent upon the pass parameters.  If the Scope does not respond to
! to the commands within 5 secs, then a call to a instruction screen
! is made so that the operator will be informed of the problem.  When
! the either a 1 or a 0 is returned from the Input_kbd sub, the Scope
! once again, is setup.
!
!************************************************************************
!****************** INTERNAL DATA DICTIONARY *****************************
!
! This Sub makes calls to the following Subs.
!***************
! Prob_inst_54110
! Input_kbd
!***************
! This Sub uses the following  Variables.
!***************
! Key_pressed   - Integer 1 or 0 used to determine which key was pressed.
!***************
! This Sub uses the following I/O Path names.
!***************
! @The54110     - I/O path for the Scope.
!
!***************** END OF INTERNAL REFERENCE SPECIFICATIONS **************
!************************************************************************
!
```

```
728   !    AT THIS TIME THE "COLOR" COMMANDS ARE NOT IMPLEMENTED
729   !
730     !
731     COM /Hp_scope/ @The54110
732     !
733   ! PRINT "Stub for Setup_54110"
734 Setup:                  !
735     ON TIMEOUT 7,5 RECOVER On_timeout
736     OUTPUT @The54110;"RESET"
737                       ! This Resets the scope to values shown on page 10-25
738                       ! in the 54110 operating manual.
739                       !
740     OUTPUT @The54110;"DISPLAY"
741     OUTPUT @The54110;"GRATICULE GRID"
742                       ! This sets up the scope  for Grid display.
743                       !
744     OUTPUT @The54110;"ACQUIRE; TYPE AVERAGE"
745     OUTPUT @The54110;"COUNT";Average
746                       ! This sets up the scope for # of Averages.
747                       !
748     OUTPUT @The54110;"TRIGGER"
749     OUTPUT @The54110;"CHANNEL 1"
750     OUTPUT @The54110;"PROBE";Atten
751                       ! The Attenuation is set for Channel 1.
752                       !
753     OFF TIMEOUT
754     GOTO End
755 On_timeout:         !
756     CALL Prob_inst_54110
757     CALL Input_kbd(Key_pressed)
758     IF Key_pressed=0 THEN Setup
759     IF Key_pressed=1 THEN Setup
760                       ! This notifies the user that there is a problem
761                       ! while trying to talk to the scope, and informs the
762                       ! user of the proper set up.
763                       !
764 End:                  !
765   SUBEND
766       !
767       !
```

```
768 Setup_3325:SUB Setup_3325(Frequency,Amplitude,Hv_onoff$)
769   !#########################################################################
770   ! SETUP  3325  SUB
771   !#########################################################################
772   ! MINI SPECIFICATIONS
773   !******************************
774   !
775   !  MODULE OVERVIEW
776   !  This module sets up the 3325 to the desired settings after presetting
777   !  it.  Pass parameters Frequency should be in Hertz,  Amplitude should
778   !  be Volts PP, and Hv_onoff$ should be either "ON" or "OFF".  Common/Hv_
779   !  mode/Hv$ is used to determine if the instrument will need to treated
780   !  as a "HV" option or not.
781   !
782   ! PROGRAMER: Murphy
783   ! LAST MOD: 4-17-86  MM
784   !           5-09-86  MM  --Added if then loop using Hv$ from Common.
785   !*************************************************************************
786   !*********************** DATA DICTIONARY ********************************
787   ! INPUTS
788   !***************
789   ! Amplitude      - Real number specifying Amplitude in VPP.
790   ! Frequency      - Real number specifying Frequency in HZ.
791   ! Hv_onoff$      - String "ON" or "OFF" to turn the High
792   !                  Voltage on or off on the 3325A.
793   ! COM/Hv_mode/Hv$ - Hv$ is "NO" if instrument is not a HV instrument.
794   !***************
795   ! OUTPUTS
796   !***************
797   ! NONE
798   !
799   !*************** END OF MINI SPECS AND DATA DICTIONARY *******************
800   !***************************************************************************
801   !
```

```
802  !************************************************************************
803  !**************** INTERNAL REFERENCE SPECIFICATIONS *********************
804  !
805  ! MODULE DISCRIPTION
806  ! This module makes calls to Subs that Preset the 3325, and then set up
807  ! the  Function,Amplitude, Frequency, and High voltage.  The Amplitude,
808  ! and Frequency will be printed on the Crt.
809  !
810  !************************************************************************
811  !********************* INTERNAL DATA DICTIONARY *************************
812  !
813  ! This Sub makes calls to the following Subs.
814  !***************
815  ! Preset_3325
816  ! Function_3325
817  ! Frequency_3325
818  ! Highvolts_3325
819  ! Amplitude_3325
820  !***************
821  ! This Sub uses NO INTERNAL Variables.
822  !***************
823  ! This Sub uses the following I/O Paths.
824  !***************
825  ! @The_3325
826  !
827  !***************** END OF INTERNAL REFERENCE SPECIFICATIONS *************
828  !************************************************************************
829      !
830    COM /Dut_3325_iopath/ @The_3325
831    COM /Hv_mode/ Hv$
832      !
833  ! PRINT "Stub for Setup_3325"
834    CALL Preset_3325(@The_3325)
835    CALL Function_3325("SQ",@The_3325)
836    CALL Frequency_3325(Frequency,@The_3325)
837    IF Hv$<>"NO" THEN
838      CALL Highvolts_3325(Hv_onoff$,@The_3325)
839    END IF
840    CALL Amplitude_3325(Amplitude,"VO",@The_3325)
841    PRINT "The Frequency and Amplitude of the waveform being measured right
now is:"
842    PRINT "FREQUENCY: ";Frequency;"HZ","AMPLITUDE: ";Amplitude;"VPP"
843  SUBEND
844      !
845      !
```

```
846 Measure:SUB Measure(Std_und_freq,Std_und_amp,Hv_und_freq,Hv_und_amp,Std_ref_
freq,Std_flat_amp,Std_flat_freqs(*),Hv_ref_freq,Hv_flat_amp,Hv_flat_freqs(*),Hv)
847    !##########################################################################
848    ! MEASURE  SUB  TEST03
849    !##########################################################################
850    ! MINI SPECIFICATIONS
851    !*******************************
852    ! MODULE OVERVIEW
853    ! This module calls the All sub which returns the first sixteen datapts
854    ! of each measurement, then calls the Undershoot sub which returns
855    ! two datapts.  Next the Amp_flatness  sub is called and returns the
856    ! number of data pts. that are specifed to be checked.  This is done
857    ! twice if the HV option is present on the 3325.  This is determined
858    ! by means of the Pass Parameter HV (0 for no HV, 1 for HV).
859    !
860    ! This sub also requires a Reference frequency be passed (Ref_freq), and
861    ! the frequency's that are to be compared to that frequency. Variables
862    ! Std_flat_freq, and Hv_flat_freq are the array of frequency's to be
863    ! compared to the Ref_freq. Any number of frequency's may be passed and
864    ! this sub will test until the last one is completed, and put the
865    ! result in Com/Users_common1/Readings_1(*).  All frequency's should be
866    ! passed in Hertz, and amplitudes in Volts PP.
867    !
868    ! PROGRAMMERS: Murphy
869    ! LAST MOD: 4-17-86
870    !**************************************************************************
871    !******************** DATA DICTIONARY ************************************
872    !
873    ! INPUTS
874    !****************
875   ! Hv             - Integer 1 or 0 for whether the instrument is a High
876   !                  votage instrument or not.
877   ! Hv_flat_amp    - Amplitude in VPP for all high voltage flatness
878   !                  measurements.
879   ! Hv_flat_freqs  - Array of frequency's in HZ for comparison with a
880   !                  reference frequency.
881   ! Hv_ref_freq    - High voltage reference frequency, used for comparison.
882   !                  Frequency in HZ.
883   ! Hv_und_freq    - Frequency in HZ used for all measurements, except
884   ! Std_flat_amp   - Amplitude in VPP for all Standard flatness measurements.
885   ! Std_flat_freqs- Array of Frequency's in HZ to be compared to a reference
886   !                  frequency's amplitude for all Standard flatness
887   !                  measurements.
888   ! Std_ref_freq   - Frequency in HZ of the Reference frequency, which all
889   !                  Standard flatness frequency amplitudes will be compared
890   !                  to.
891   ! Std_und_amp    - Amplitude in VPP for all measurement, other than
892   !                  flatness, during High voltage measurements.
893   ! Hv_und_amp     - Voltage in VPP for all measurements, except flatness
894   !                  during High voltage measurements.
895   ! Std_und_freq   - Frequency in HZ of all measurements, other than
896   !                  Flatness for Standard measurements.
897   !*****************
898   ! OUTPUTS
899   !*****************
900   ! Readings_1     - 45 measurement readings put in Common "User_common1".
901   ! U$             - Not used.
902   !
903   !********************* END OF MINI SPECS AND DATA DICTIONARY **************
```

```
!*****************************************************************************
!
```

```
906   !***************************************************************************
907   !******************* INTERNAL REFERENCE SPECIFICATIONS *****************
908   !
909   ! MODULE DISCRIPTION
910   ! See Module Overview (above).
911   !
912   !**************************************************************************
913   !******************* INTERNAL DATA DICTIONARY ***************************
914   !
915    ! This Sub makes calls to the following Subs.
916    !***************
917    ! Check_dut
918    ! Hv_instruct_1
919    ! Input_kbd
920    ! Setup_3325
921    ! Autoscle_54110
922    ! Hv_instruct_2
923    ! All_54110
924    ! Undersht_54110
925    ! Amp_flatness
926    ! Format
927    !***************
928    ! This Sub uses the following Variables Internally
929   !***************
930   ! Amplitude      - Amplitude in Vpp for setting up 3325.
931   ! Array_ele      - Used to keep track of Array elements, (a counter).
932   ! Base           - The most common voltage value of the -cycle of
933   !                  the wave.  This is obtained from the Base measurement
934   !                  from the Scope All function and retrieved through Com.
935   ! Channel        - Integer 1 or 2, specifying scope Channel.
936   ! F_readings     - An array of Measurements returned from the Flatness
937   !                  measurement Sub.
938   ! Flatness_amp   - The amplitude in VPP passed to the Flatness measurement
939   !                  Sub.
940   ! Flatness_ref   - The Reference frequency passed to the Flatness
941   !                  measurement Sub.
942   ! Freq_counter   - Integer counter used to determine which setup(Frequency,
943   !                  Amplitude, High Voltage, etc.) for measurements.
944   ! Frequency      - Frequency in HZ for setting up the 3325.
945   ! Frequencys     - Variable array, passed to the Flatness measurement Sub,
946   !                  of the Frequency's to be measured.
947   ! J              - Integer Counter.
948   ! Key_pressed    - Integer 1 or 0 used to determine if the "Enter" or
949   !                  "Space" key were pressed.
950   ! No_of_freadings-Integer number used to keep track of how many Flatness
951   !                  readings are to be made.
952   ! No_of_times    - Integer 1 or 2, indicating Number of times test is to
953   !                  be looped (depends if the instrument is a High Voltage
954   !                  option instrument or not.
955   ! Offset         - Integer used to keep track of array elements depending
956   !                  if the No_of_times is 1 or 2.
957   ! Period         - A time in nSec, of one cycle of the waveform
958   ! Readings       - An array used as a manipulating and temporary holding
959   !                  of measurements before being put into Com "Readings_1".
960   ! Top            - The most common voltage value of the +cycle of
961   !                  the wave.  This is obtained from the Base measurement
962   !                  from the Scope All function and retrieved through Com.
963   ! Topbase        - The sum of the absolute value of the Top and Base
964   !                  measurements.
```

```
965   ! U_readings    - An array of 2, that is passed back from the Undershoot
966   !                 measurement Sub as the Undershoot measurments.
967   ! Hv_onoff$     - A string passed to the Setup_3325 sub, indicating
968   !                 if the High voltage should be "on or "off".
969   !
970   !************* END OF INTERNAL REFERENCE SPECIFICATIONS *****************
971   !***********************************************************************
972   !
```

```
973      !
974     COM /User_common1/ Readings_1(*),U$
975      !
976      !   PRINT "Stub for Measure"
977                         !
978     Offset=0
979           ! Determines offset needed depending upon if HV is
980           ! being measured.
981           !
982     Freq_counter=0
983                           ! Initalizes counter that is used whether the HV
984                           ! option is present.
985                           !
986     No_of_times=Hv+1
987                           ! If Hv option is present No_of_times = 2,
988                           ! or if no Hv then No_of_times = 1.
989                           !
990     CALL Check_dut
991                    ! Makes sure Dut is OK.
992                    !
993     FOR Freq_counter=1 TO No_of_times
994                    !
995       ALLOCATE F_readings(1:10)
996       ALLOCATE Frequencys(1:10)
997                    ! Allocates enough room for up to 10, Flatness readings
998                    ! (F_readings(*)) and Flatness frequency's to be
999                    ! passed to the Amplitude flatness sub (Frequencys(*)).
1000                   !
1001      SELECT Freq_counter
1003      CASE 1
1004        Frequency=Std_und_freq
1005        Amplitude=Std_und_amp
1006        Hv_onoff$="off"
1007        Flatness_ref=Std_ref_freq
1008        Flatness_amp=Std_flat_amp
1009        Channel=1
1010        J=1
1011        WHILE Std_flat_freqs(J)>0
1012          No_of_freadings=J
1013          Frequencys(J)=Std_flat_freqs(J)
1014          J=J+1
1015        END WHILE
1016                          ! Determines number of Flatness frequency's
1017                          ! to be measured, and assigns values to
1018                          ! Frequencys variable.
1019                          !
1020      CASE 2
1021                          ! Set up for NO Hv option
1022                          ! measurements.
1023                          !
1025        Frequency=Hv_und_freq
1026        Amplitude=Hv_und_amp
1027        Hv_onoff$="on"
1028        Flatness_ref=Hv_ref_freq
1029        Flatness_amp=Hv_flat_amp
1030        Channel=1
1031        J=1
1032        WHILE Hv_flat_freqs(J)>0
1033          No_of_freadings=J
```

```
1034          Frequencys(J)=Hv_flat_freqs(J)
1035          J=J+1
1036        END WHILE
1037                          ! Determines number of Flatness frequency's
1038                          ! to be measured, and assigns values to
1039                          ! Frequencys variable.
1040                          !
1041      CASE ELSE
1042        OUTPUT CRT;"PROGRAM HAS ENCOUNTERED AN UNEXPECTED LOOP VALUE"
1043        OUTPUT CRT;"PLEASE CALL YOUR SYSTEM ADMINISTRATOR AND INFORM"
1044        OUTPUT CRT;"HER THAT THE 'MEASURE' SUB HAS A PROBLEM."
1045      END SELECT
1047                                     ! Set up for HV option
1048                                     ! measurements.
1049                                     !
1050      DIM Readings(1:24)
1051      DIM U_readings(1:3)
1052 Check_routine:        !
1053      All_set=1                  !Initializw Boolean variable
1055      REPEAT
1056        IF Freq_counter=2 THEN
1057          REPEAT
1059            BEEP
1060            CALL Hv_instruct_1
1061            CALL Input_kbd(Key_pressed)
1062          UNTIL Key_pressed=1
1065        END IF
1066        CALL Setup_3325(Frequency,Amplitude,Hv_onoff$)
1067        CALL Autoscle_54110
1068        IF Freq_counter=2 THEN
1069        ! Manual_54110(Frequency/2,Amplitude/3,Channel)
1070                                   ! Manual_54110 is a little bit faster
1071                                   ! then Autoscle.
1072          All_set=0
1074          CALL Hv_instruct_2
1075          CALL Input_kbd(Key_pressed)
1076          IF Key_pressed=1 THEN All_set=1
1077                                    ! The input_kbd returns a 0 if
1078                                    ! the Return key is pressed or a
1079                                    ! 1 if the space bar is pressed.
1080                                    !
1081        END IF
1082      UNTIL All_set=1
1084                                    ! This part informs the operator to
1085                                    ! the connections neccesary for the
1086                                    ! HV measurements, and lets them look
1087                                    ! at the signal on the scope.
1088      CALL All_54110(Readings(*))
1089                                    !
1090                                    !
1091                                    !
1092      Period=Readings(2)             ! Readings 2 is the period
1093                                     ! from the All measurement.
1094                                     !
1095      Topbase=Readings(7)            ! Readings 7 is the TopBase
1096                                     ! from the All measurement.
1097                                     !
1098      Top=Readings(15)               ! Readings 15 is the Top
1099                                     ! from the All measurement.
1100                                     !
```

```
1101      Base=Readings(16)              ! Readings 16 is the Base
1102                                     ! from the All measurement.
1103                                     !
1104      CALL Undersht_54110(Period,Top,Base,U_readings(*))
1105           !
1106      FOR Array_ele=17 TO 18
1107        Readings(Array_ele)=U_readings(Array_ele-16)
1108      NEXT Array_ele
1109                   ! Fills Readings Array with Undershoot measurements.
1110                   !
1111      CALL Amp_flatness(Flatness_ref,Frequencys(*),Flatness_amp,F_readings(*
))
1112                   !
1113      FOR Array_ele=19 TO 19+No_of_freadings-1
1114!          PRINTER IS 701       !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
1115!          PRINT "Array_ele=";Array_ele,"No_of_freadings=";No_of_freadings,"
F_readings(Array_ele-18)";F_readings(Array_ele-18)
1116!          PRINTER IS 1         !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
1117        Readings(Array_ele)=F_readings(Array_ele-18)
1118      NEXT Array_ele
1119                   ! Fills Readings array with Flatness measurements.
1120                   !
1121      CALL Format(Readings(*),24)
1122                   !
1123      FOR Array_ele=1 TO 18+No_of_freadings
1124        Readings_1(Array_ele+Offset)=Readings(Array_ele)
1125      NEXT Array_ele
1126                   ! Fills Com Readings_1 array with 18 readings.
1127                   !
1128      Offset=Offset+18+No_of_freadings
1129                   ! Sets Offset to value inwhich to offset each reading
1130                   ! for the next 18 sets of measurements.
1131                   !
1132      DEALLOCATE F_readings(*)
1133      DEALLOCATE Frequencys(*)
1134                   ! By DE-Allocating the array's are reset to zero
1135                   ! so there will be no confusion in the data being
1136                   ! passed back from Amp flatness Sub.
1137                   !
1138    NEXT Freq_counter
1139                   !
1140  SUBEND           ! End of Measure Sub program.
1141    !
1142    !
```

```
1143 Cleanup:SUB Cleanup  !
1144  !#########################################################################
1145  ! CLEANUP  SUB  PROGRAM
1146  !#########################################################################
1147  ! MINI SPECIFICATIONS
1148  !*******************************
1149  ! MODULE OVERVIEW
1150  ! This sub Cleanup is the last sub called before all of the datapts are
1151  ! passed to Executive.
1152  !*************************************************************************
1153  !******************** DATA DICTIONARY ************************************
1154  !INPUTS
1155  !**************
1156  ! NONE
1157  !**************
1158  ! OUTPUTS
1159  !**************
1160  ! NONE
1161  !
1162  !***************** END OF MINI SPECS AND DATA DICTIONARY *****************
1163  !*************************************************************************
1164  !
```

```
1165  !*************************************************************************
1166  !***************** INTERNAL REFERENCE SPECIFICATIONS *********************
1167  ! MODULE DISCRIPTION
1168  ! See Module Overview (above).
1169  !
1170  !*************************************************************************
1171  !******************* INTERNAL DATA DICTIONARY ****************************
1172  !
1173  ! This Sub calls the following Subs.
1174  !**************
1175  ! Preset_3325
1176  !**************
1177  ! This Sub uses the following I/O Paths.
1178  !**************
1179  ! @The 54110    - I/O path for the Scope.
1180  ! @The_3325     - I/o path for the 3325 as a DUT.
1181  !
1182  !************** END OF INTERNAL SPECIFICATIONS ***************************
1183  !*************************************************************************
1184  !
```

```
1185  !
1186    COM /Hp_scope/ @The54110
1187    COM /Dut_3325_iopath/ @The_3325
1188  !
1189    CALL Preset_3325(@The_3325)
1190                              ! Set 3325 back to Preset state.
1191    OUTPUT KBD USING "#,K";" K"
1192    !
1193    !
1194  SUBEND
1195    !
1196    !
```

```
1197 Format:SUB Format(Readings(*),Total_readings)
1198   !########################################################################
1199   ! FORMAT  SUB PROGRAM
1200   !########################################################################
1201   ! MINI SPECIFICATIONS
1202   !*****************************
1203   ! MODULE OVERVIEW
1204   ! This Module allows the programmer to modify the format of Readings(*).
1205   ! This sub must also be passed a the number of readings to be Formatted
1206   ! by means of the variable, Total_readings.
1207   !************************************************************************
1208   !******************* INTERNAL DATA DICTIONARY ***************************
1209   !
1210   ! INPUTS
1211   !**************
1212   ! Readings     - Array of measurements too be formated.
1213   ! Total_readings- Total number of readings to be formated.
1214   !**************
1215   ! OUTPUTS
1216   !**************
1217   ! Readings     - Array of formated measurements in the same order as they
1218   !                were passed into the Sub.
1219   !
1220   !************* END OF MINI SPECS AND DATA DICTIONARY ********************
1221   !************************************************************************
1222   !
1223   !************************************************************************
1224   !****************** INTERNAL SPECIFICATIONS *****************************
1225   ! MODULE SPECIFICATIONS
1226   ! See Module Overview (above).
1227   !
1228   !************************************************************************
1229   !******************* INTERNAL DATA DICTIONARY ***************************
1230   !
1231   ! This Sub makes NO CALLS to any other Subs.
1232   !**************
1233   ! This Sub uses the following variables internally.
1234   !**************
1235   ! Counter      - Integer used to count.
1236   !
1237   !***************END OF INTERNAL REFERENCE SPECIFICATIONS ****************
1238   !************************************************************************
1239   !
```

```
1240    Counter=0
1241                ! Initializes Counter
1242                !
1243    Readings(1)=Readings(1)/10^3
1244                ! Frequency in KHZ.
1245                !
1246    Readings(2)=Readings(2)*10^9
1247                ! Period in nSec.
1248                !
1249    Readings(3)=Readings(3)*10^9
1250                ! Positive width in nSec.
1251                !
1252    Readings(4)=Readings(4)*10^9
1253                ! Negative width in nSec.
1254                !
1255    Readings(5)=Readings(5)*10^9
1256                ! Risetime in nSec.
1257                !
1258    Readings(6)=Readings(6)*10^9
1259                ! Falltime in nSec.
1260                !
1261    Readings(9)=Readings(9)/Readings(7)*100
1262                ! Falltime Overshoot in %, rounded off.
1263                !
1264    Readings(10)=Readings(10)/Readings(7)*100
1265                ! Risetime Overshoot in %, rounded off.
1266                !
1267    Readings(17)=Readings(17)/Readings(7)*100
1268                ! Risetime Undershoot in %, rounded off.
1269                !
1270    Readings(18)=Readings(18)/Readings(7)*100
1271                ! Falltime Undershoot in %, rounded off.
1272                !
1273    FOR Counter=1 TO Total_readings
1274      Readings(Counter)=DROUND(Readings(Counter),2)
1275    NEXT Counter
1276                ! This rounds off all readings to two places.
1277                !
1278     !
1279  SUBEND                      ! End of Format
1280     !
1281     !
```

```
1282 Tweak_dut:SUB Tweak_dut(Frequency,Amplitude,Hv_option)
1283     !######################################################################
1284     ! TWEAK DUT  SUB  3325A
1285     !######################################################################
1286     ! MINI SPECIFICATIONS
1287     !****************************
1288     !
1289     !  MODULE OVERVIEW
1290     ! This sets up 3325A and the scope, so that the operator may view
1291     ! waveform and Tweak if desired.  This SUB must be passed a Frequency
1292     ! and Amplitude so that the 3325A may be set to the desired tweak
1293     ! settings.  This SUB will also determine if the instrument is a HV
1294     ! option instrument or not and return a value of 1 (if present) or 0
1295     ! for the value of variable Hv_option.  Amplitude should be in Volts PP
1296     ! and Frequency in Hertz.
1297     !
1298     ! PROGRAMER: Murphy
1299     ! LAST MOD: 4-17-86
1300     !**********************************************************************
1301     !************************* DATA DICTIONARY ****************************
1302     !
1303     ! INPUTS
1304     !***************
1305     ! Amplitude     - Amplitude in VPP.
1306     ! Frequency     - Frequency in HZ.
1307     !***************
1308     ! OUTPUTS
1309     !***************
1310     ! Hv_option     - An integer 1 or 0, a 1 is passed if the instrument
1311     !                 is a High Voltage option instrument, otherwise a 0.
1312     !
1313     !************** END OF MINI SPECS AND DATA DICTIONARY *****************
1314     !**********************************************************************
1315     !
1316     !**********************************************************************
1317     !**************** INTERNAL REFERENCE SPECIFICATIONS *******************
1318     !
1319     ! MODULE SPECIFICATIONS
1320     ! This module calls one of three instruction subs depending on the
1321     ! circumstances. If the 3325 does not respond within 5 secs. the
1322     ! Connect_inst is called informing the operator of the proper
1323     ! connection precedure. Once the instrument is connected, the
1324     ! instructions are displayed to the operator that are relevent
1325     ! to the type of instrument (HV or Standard).  If the Space bar
1326     ! is pressed the sub is exited.
1327     !**********************************************************************
1328     !******************* INTERNAL DATA DICTIONARY *************************
1329     !
1330     ! This Sub makes calls to the following Subs.
1331     !***************
1332     ! Hv_present_3325
1333     ! Setup_3325
1334     ! Manual_54110
1335     ! Hv_present_inst
1336     ! Std_inst
1337     ! Connnect_inst
1338     !***************
1339     ! This Sub makes use of the following variables internally.
1340     !***************
```

```
1341     ! Channel         - Integer 1 or 2, specifying the desired channel.
1342     ! Half_frequency- Frequency variable divided by 2.
1343     ! Hv$             - String used to specify in Common under the name
1344     !                   "Hv_mode" indicating if the instrument is a HV
1345     !                   option or not.
1346     ! Hv_onoff$       - String specifying if the High voltage option
1347     !                   should be turned on or off.
1348     ! Hv_or_rf$       - A string returned by the Hv_present sub indicating
1349     !                   if the instrument is a HV or Front rear instrument.
1350     !
1351     !***************
1352     ! The following I/O paths are used.
1353     !***************
1354     ! @The_3325       - I/O path for the 3325A as a DUT.
1355     !
1356     !*************** END OF INTERNAL REFERENCE SPECIFICATIONS *************
1357     !*********************************************************************
1358     !
```

```
1359      !
1360    COM /Hv_mode/ Hv$
1361    COM /Dut_3325_iopath/ @The_3325
1362    Channel=1
1363    Hv_onoff$="off"
1364                  !
1365    Half_freq=Frequency/2
1366                  ! Makes it so that two cycles are displayed on
1367                  ! scope.
1368    ON TIMEOUT 7,2 RECOVER On_timeout
1369    CLEAR @The_3325
1370    CALL Hv_present_3325(Hv_or_fr$,@The_3325)
1371    CALL Setup_3325(Frequency,Amplitude,Hv_onoff$)
1372    CALL Manual_54110(Half_freq,Amplitude,Channel)
1373   !   PRINT Hv_or_fr$
1374   !   PAUSE
1375    IF Hv_or_fr$="HV" THEN
1376      CALL Hv_present_inst
1377      Hv_option=1
1378    ELSE
1379      CALL Std_inst
1380      Hv_option=0
1381      Hv$="NO"
1382                     ! Sets common Hv$ = NO so that proper datapts
1383                     ! will be eliminated.  this eliminates the
1384                     ! need for the operator to select the HV option.
1385                     !
1386    END IF
1387    OFF TIMEOUT
1388                     ! Turns TIMEOUT interrupt off.
1389                     !
1390    GOTO End
1391                     ! Skips over Connection instruction.
1392                     !
1393 On_timeout:     !
1394    CALL Connect_inst
1395    OFF TIMEOUT
1396                     ! Prints Connection instructions if timeout occurs.
1397                     !
1398 End: !
1399  SUBEND             ! End of Tweak_dut SUB.
1400       !
1401       !
```

```
1402 Input_kbd:SUB Input_kbd(Key_pressed)
1403   !#######################################################################
1404   ! INPUT  KEYBOARD  SUB  OPERATOR  INTERFACE
1405   !#######################################################################
1406   ! MINI SPECIFICATIONS
1407   !******************************
1408   ! MODULE OVERVIEW
1409   ! This sub checks if the Space Bar or Return key are pressed.  If
1410   ! any other key is pressed the sub waits.  The result of the key
1411   ! that was pressed is then passed back to the Caller by the Key_
1412   ! pressed parameter. Space Bar =1. Return key = 0.
1413   !
1414   ! PROGRAMER: Murphy
1415   ! LAST MOD: 4-17-86    MM
1416   !***********************************************************************
1417   !******************** DATA DICTIONARY ***********************************
1418   ! INPUT
1419   !****************
1420   ! NONE
1421   !****************
1422   ! OUTPUT
1423   !****************
1424   ! Key_pressed    - An integer 1 or 0, 1= Space Bar  0= Return Key.
1425   !
1426   !******************* END OF MINI SPECS AND DATA DICTIONARY **************
1427   !***********************************************************************
1428   !
1429   !***********************************************************************
1430   !******************* INTERNAL REFERENCE SPECIFICATIONS ******************
1431   ! MODULE SPECIFICATIONS
1432   ! See Module Overview (above).
1433   !***********************************************************************
1434   !******************** INTERNAL DATA DICTIONARY **************************
1435   !
1436   ! This Sub makes NO CALLS to any other Subs.
1437   !****************
1438   ! The following is a list of variables used internally.
1439   !****************
1440   ! Sys_prio       - System Priority is a integer equal to the valu
1441   !                  of the priority level of the system in this sub.
1442   ! Key$           - String indicating what key was pressed.
1443   ! Sys_prio$      - String used to hold the what the system priority
1444   !                  level is in this sub.
1445   !
1446   !***************** END OF INTERNAL REFERENCE SPECIFICATIONS *************
1447   !***********************************************************************
1448   !
```

```
1449     !
1450    Sys_prio$=SYSTEM$("SYSTEM PRIORITY")
1451    IF VAL(Sys_prio$)<15 THEN
1452      Sys_prio=VAL(Sys_prio$)+1
1453    ELSE
1454      DISP " ERROR in Input_kbd SUB, Interupt level > 15."
1455    END IF
1456             ! This part makes sure that the ON KBD will have a greater
1457             ! priority level then the SUB itself.
1458             !
1459 On_kbd: !
1460    ON KBD,Sys_prio GOTO Read_keys
1461 Wait:GOTO Wait
1462               !
1463 Read_keys: !
1464    Key$=KBD$
1465    OFF KBD
1466              ! Turns keyboard check off to garantee no interupts while in
1467              ! Case Select process.
1468              !
1469    SELECT Key$
1470    CASE " "
1471      Key_pressed=1
1472               ! If Space bar pressed
1473    CASE " E"
1474      Key_pressed=0
1475               ! If Return key pressed
1476    CASE ELSE
1477      GOTO On_kbd
1478    END SELECT
1479               !
1480    OUTPUT KBD USING "#,K";" K"
1481               ! Clears screen
1482               !
1483  SUBEND       ! END OF SUB INPUT_KBD
1484       !
1485       !
```

```
1486 Amp_flatness:SUB Amp_flatness(Flatness_ref,Frequencys(*),Flatness_amp,F_rea
dings(*))
1487  !##########################################################################
1488  ! AMPLITUDE  FLATNESS   SUB
1489  !##########################################################################
1490  ! MINI SPECIFICATIONS
1491  !*******************************
1492  !
1493  ! MODULE OVERVIEW
1494  ! This sub will use the variables Flatness_ref (freq), and Flatness_amp as
1495  ! references and then use the variable array Frequencys as the
1496  ! frequency's to be compared to the reference.  The results will be
1497  ! passed back using the array F_readings.  The results will be passed
1498  ! back in a % format using the Topbase measurements from the scope.
1499  ! Frequency's should be in Hertz and Amplitudes in Volts PP.
1500  !
1501  ! PROGRAMER: MURPHY MOORE
1502  ! LAST MOD 4-17-86  MM
1503  !*************************************************************************
1504  !*********************** DATA DICTIONARY *********************************
1505  !
1506  ! INPUTS
1507  !****************
1508  ! Flatness_ref    - Frequency in HZ of the Reference frequency that
1509  !                   all other frequency's amplitude is to be compared
1510  !                   to.
1511  ! Frequencys      - Array of frequency's in HZ that the amplitudes must
1512  !                   be compared to the Reference frequency's amplitude.
1513  ! Flatness_amp    - Amplitude in Vpp that will be used for all flatness
1514  !                   measurements.
1515  !****************
1516  ! OUTPUTS
1517  !****************
1518  ! F_readings      - Array of readings in percentage form in deviation from
1519  !                   the reference frequency's common mode voltages(Topbase)
1520  !
1521  !***************END OF MINI SPECS AND DATA DICTIONARY *******************
1522  !*************************************************************************
1523  !
1524  !*************************************************************************
1525  !**************** INTERNAL REFERENCE SPECIFICATIONS *********************
1526  ! MODULE DISCRIPTION
1527  ! This module measures the sum of the most common voltage of the +cycle
1528  ! and the -cycle of the Reference Frequency's amplitude useing the sopes
1529  ! Topbase measurement. It then does the same for the  Frequency's to be
1530  ! measured and puts the diference in a precentage format in the F_readings
1531  ! array.
1532  !*************************************************************************
1533  !******************* INTERNAL DATA DICTIONARY ***************************
1534  !
1535  ! This Sub makes calls to the following Subs.
1536  !****************
1537  ! Frequency_3325
1538  ! Amplitude_3325
1539  ! Autoscle_54110
1540  ! Manual_54110
1541  !****************
1542  ! This Sub uses the following variables internally.
1543  !****************
```

```
1544  ! Channel         - Integer 1 or 2, indicating the desired channel of the
1545  !                   scope to be used.
1546  ! I               - Used as a counter to count the number of frequency's to
1547  !                   be measured.
1548  ! Ref_reading     - The Topbase reading of the Reference Frequency in Volts
1549  ! Temp_ readings  - A temporary variable used to hold the readings of each
1550  !                   of the Frequency's Topbase measurement.
1551  ! Unit$           - Used to indicate units for the frequency when the
1552  !                   3325 Sub is called ("VO" indicates VPP).
1553  !****************
1554  ! The following I/O Paths are used.
1555  !****************
1556  ! @The54110       - Scope I/O path.
1557  ! @The_3325       - 3325A I/O path.
1558  !
1559  !***************** END OF INTERNAL REFERENCE SPECIFICATIONS **************
1560  !**************************************************************************
1561  !
```

```
1562    !
1563    COM /Hp_scope/ @The54110
1564    COM /Dut_3325_iopath/ @The_3325
1565    !
1566    DISP "Making flatness measurements........"
1567                     ! Notify's user of measurements being made.
1568    Channel=1
1569    Unit$="VO"
1570    CALL Frequency_3325(Flatness_ref,@The_3325)
1571    CALL Amplitude_3325(Flatness_amp,Unit$,@The_3325)
1572    CALL Autoscle_54110
1573    OUTPUT @The54110;"MEASURE"
1574    OUTPUT @The54110;"TOPBASE?"
1575    ENTER @The54110 USING "MD.5DESZZ,2X";Ref_reading
1576                     ! This sets up and measures the reference frequency.
1577                     !
1578    I=1              ! Initializes counter.
1579    WHILE Frequencys(I)>0
1580!     PRINTER IS 701             !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
1581!     PRINT "Frequencys(I)=";Frequencys(I),"I=";I
1582!     PRINTER IS 1               !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
1583      CALL Frequency_3325(Frequencys(I),@The_3325)
1584      CALL Manual_54110(Frequencys(I)/2,10,Channel)
1585      OUTPUT @The54110;"MEASURE"
1586      OUTPUT @The54110;"TOPBASE?"
1587      ENTER @The54110 USING "MD.5DESZZ,2X";Temp_readings
1588!       PRINTER IS 701           !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
1589!       PRINT "Ref_reading=";Ref_reading,"Temp_readings=";Temp_readings
1590      F_readings(I)=(Temp_readings-Ref_reading)/Ref_reading*100
1591!      PRINTER IS 1              !<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
1592                     !
1593      I=I+1
1594    END WHILE
1595                     ! This loop keeps measureing the V Topbase of
1596                     ! of each frequency and storeing them in the array
1597                     ! F_readings until there are no more.
1598                     !
1599    DISP ""
1600                     ! Clears display line.
1601  SUBEND                            ! End of Amplitude flatness
1602    !
1603    !
```

```
1604 Hv_instruct_2:SUB Hv_instruct_2
1605   !############################################################################
1606   ! HIGH  VOLTAGE OPTION INSTRUCTIONS   SUB   DISPLAY FOR OPERATOR
1607   !############################################################################
1608   ! MODULE OVERVIEW
1609   ! This sub instructs the operator that the HV signal should be seen
1610   ! on the scope and, in order to continue the space bar may  be pressed.
1611   !****************************************************************************
1612   !
1613   ! This Sub makes NO CALLS to any other Subs.
1614   !***************
1615     !
1616     OUTPUT KBD USING "#,K";" K"
1617                                ! Clears screen.
1618                                !
1619     PRINT TABXY(1,2);"You should be able to see a 3VPP, 550KHZ square wave o
n the scope."
1620     PRINT TABXY(1,6);"IF NOT  PRESS  the  [RETURN]  key."
1621     PRINT TABXY(1,8);"PRESS  the  [SPACE BAR]  if the signal looks OK."
1622     PRINT TABXY(1,10);"NOTE: After the Scope is done making it's measurement
s the 3325 may be"
1623     PRINT TABXY(7,11);"disconnected, while the readings are being passed to
Executive."
1624        !
1625        !
1626  SUBEND                                  ! End of Hv_instructions_2
1627       !
1628       !
```

```
1629 Connect_inst:SUB Connect_inst
1630   !########################################################################
1631   !  CONNECTION INSTRUCTIONS FOR 3325   SUB   DISPLAY OPERATOR INSTRUCTIONS
1632   !########################################################################
1633   ! MODULE OVERVIEW
1634   ! This module displays instructions to the operator on how to connect
1635   ! the 3325 to the HP Scope.
1636   !************************************************************************
1637   !
1638   ! this Sub makes NO CALLS to any other Subs.
1639   !***************
1640     !
1641    OUTPUT KBD USING "#,K";" K"
1642                                 ! Clears screen.
1643                                 !
1644    PRINT TABXY(1,1);"(1)"
1645    PRINT TABXY(4,2);"Make sure the 3325A is turned on, the addres is set to
 ""17"""
1646    PRINT TABXY(4,3);"and the HP-IB cable is connected."
1647    PRINT TABXY(1,4);"(2)"
1648    PRINT TABXY(4,5);"Please connect a ""6db PAD"" to the  ""OUTPUT""  of th
e 3325A."
1649    PRINT TABXY(1,7);"(3)"
1650    PRINT TABXY(4,8);"Connect a ""BNC CABLE"" from the divider to the  ""INP
UT 1"""
1651    PRINT TABXY(4,9);"of the HP SCOPE."
1652    PRINT TABXY(1,11);"PRESS  [RETURN] or [ENTER]  key if you would like to
verify the"
1653    PRINT TABXY(8,12);"Wave, or Tweak the instrument."
1654    PRINT TABXY(1,14);"PRESS  [SPACE BAR]  if you would like to start making
 measurements."
1655    PRINT TABXY(1,16);"REMINDER:  If you press the  [SPACE BAR]  measureing
will begin IMMEDIATELY."
1656    PRINT TABXY(15,18);">>>>>>>>>>>>> BYE <<<<<<<<<<<<<<"
1657   !
1658   !
1659  SUBEND                               ! End of connect_instrument
1660     !
1661     !
```

```
1662 Std_inst:SUB Std_inst
1663   !########################################################################
1664   ! STANDARD INSTRUMENT INSTRUCTIONS  SUB  DISPLAY  OPERATOR INSTRUCTIONS
1665   !########################################################################
1666   ! MODULE OVERVIEW
1667   ! Displays instructions to the operator and informs them that this
1668   ! instrument is a standard instrument with Rear Front connections.
1669   !************************************************************************
1670   !
1671   ! This Sub makes NO CALLS to any other Subs.
1672   !***************
1673     !
1674    OUTPUT KBD USING "#,K";" K"
1675    PRINT TABXY(1,1);">>>>>>>>>>>>>>>>>>> STANDARD INSTRUMENT --NO HV <<<<<<
<<<<<<<<<<<<<"
1676    PRINT TABXY(1,2);"LOOKS GOOD!"
1677    PRINT TABXY(1,3);"As far as I can tell, you have the 3325A  Plugged in,
HP-IB connected,"
1678    PRINT TABXY(1,4);"power on, and the address set to 17.  If you're really
 on top of things,"
1679    PRINT TABXY(1,5);"you have connected  a ""2:1 DIVIDER"" to the ""OUTPUT"
"of the 3325A,"
1680    PRINT TABXY(1,6);"and a ""BNC CABLE"" going to the ""INPUT 1"" on the 54
110 scope."
1681    PRINT TABXY(1,7);"If you haven't, you need to do so now."
1682    PRINT TABXY(1,9);"Now there should be a 1.1 MHz square wave present on t
he Scope."
1683    PRINT TABXY(1,10);"NOTICE: This instrument will be tested as a ""STANDAR
D"" instrument."
1684    PRINT TABXY(9,11);"If INCORRECT--Leave instrument connected, and PRESS t
he [RETURN] key."
1685    PRINT TABXY(9,12);"IF CORRECT----PRESS the [SPACE BAR] to begin testing
"
1686    PRINT TABXY(1,14);"REMEMBER   Testing will begin immediatly after the [S
PACE BAR] is pressed."
1687    PRINT TABXY(13,16);">>>>>>>>>>>>>> HAVE A NICE TEST <<<<<<<<<<<<<<<<<"
1688                 !
1689                 !
1690  SUBEND         ! End of Std_inst
1691     !
1692     !
```

```
1693 Hv_present_inst:SUB Hv_present_inst
1694   !#######################################################################
1695   ! HIGH VOLTAGE PRESENT INSTRUCTIONS  SUB   DISPLAY  OPERATOR INSTRUCTIONS
1696   !#######################################################################
1697   ! MODULE OVERVIEW
1698   ! Displays instructions to the operator and informs them that this
1699   ! instrument is a High Voltage option instrument.
1700   !***********************************************************************
1701   !
1702   ! This Sub makes NO CALLS to any other Subs.
1703   !***************
1704     !
1705    OUTPUT KBD USING "#,K";" K"
1706    PRINT TABXY(1,1);">>>>>>>>>>>>>>>>>>>>>>>> HV OPTION INSTRUMENT <<<<<<<
<<<<<<<<<<<<<<"
1707    PRINT TABXY(1,2);"LOOKS GOOD!"
1708    PRINT TABXY(1,3);"As far as I can tell, you have the 3325A  Plugged in,
HP-IB connected,"
1709    PRINT TABXY(1,4);"power on, and the address set to 17.  If you're really
 on top of"
1710    PRINT TABXY(1,5);"things, you have connected  a ""2:1 DIVIDER"" to the "
"OUTPUT""of the 3325A"
1711    PRINT TABXY(1,6);"with a ""BNC CABLE"" going to the ""INPUT 1"" on the 5
4110 scope."
1712    PRINT TABXY(1,7);"If you haven't, you need to do so now."
1713    PRINT TABXY(1,9);"Now there should be a 1.1 MHz square wave present on t
he Scope."
1714    PRINT TABXY(1,10);"NOTICE: This instrument will be tested as a ""HIGH VO
LTAGE"" option instrument."
1715    PRINT TABXY(9,11);"If INCORRECT--Leave instrument connected, and PRESS t
he [RETURN] key."
1716    PRINT TABXY(9,12);"If CORRECT----PRESS the [SPACE BAR] to begin testing
"
1717    PRINT TABXY(1,14);"REMEMBER   Testing will begin immediatly after the [S
PACE BAR] is pressed."
1718    PRINT TABXY(15,17);">>>>>>>>>>>>> HAVE A NICE TEST <<<<<<<<<<<<<<<"
1719                   !
1720                   !
1721  SUBEND           ! End of Std_inst
1722     !
1723     !
```

```
1724 Hv_instruct_1:SUB Hv_instruct_1
1725  !#########################################################################
1726  ! HIGH  VOLTAGE OPTION INSTRUCTIONS   SUB   DISPLAY FOR OPERATOR
1727  !#########################################################################
1728  ! MODULE OVERVIEW
1729  ! This sub displays to the user the instructions for connecting and
1730  ! instrument with a HV option to the scope.
1731  !*************************************************************************
1732  !
1733  ! This Sub makes NO CALLS to any other Subs.
1734  !****************
1735    !
1736    OUTPUT KBD USING "#,K";" K"
1737                               ! Clears screen.
1738                               !
1739    PRINT TABXY(1,2);"You have now gotten to the High Voltage Option, portio
n of the test."
1740    PRINT TABXY(1,3);"Please connect the  ""10:1 Divider""  to the  ""INPUT
 1"", on the Scope (54110)"
1741    PRINT TABXY(1,8);"PRESS  the  [SPACE BAR]  in order to view the signal,
and to continue"
1742    PRINT TABXY(26,9);"the testing."
1743       !
1744       !
1745  SUBEND                                ! End of Hv_instructions_1
1746      !
1747      !
```

IEEE 488 FUNCTIONAL ASPECTS

PRELUDE - THE IEEE 488-1975 STANDARD IS IMPLEMENTED IN A BYTE SERIAL, BIT PARALLEL INTERFACE ARRANGEMENT, AND IS PRIMARILY INTENDED FOR PROGRAMMABLE TEST AND MEASUREMENT INSTRUMENTATION SYSTEMS. THE TERMINOLOGY BYTE SERIAL, BIT PARALLEL MIGHT SEEM CONFUSING AT FIRST, UNTIL ONE SEES HOW INFORMATION IS SENT OVER THE BUS. AN 8 LINE WIDE DATA BUS IS USED. THIS CORRESPONDS TO BYTE WIDE INFORMATION TRANSFER. MOST OFTEN ASCII CHARACTER REPRESENTATION (7 BIT) IS USED. EACH GROUP OF 8 BITS (OR ONE ASCII CHARACTER) IS SENT AT ONCE OR IN PARALLEL (BIT PARALLEL). USUALLY MESSAGES CONSIST OF 10 TO 20 CHARACTERS (OR BYTES). EACH IS SENT ONE AFTER THE OTHER OR SERIALLY (BYTE SERIAL).

KEY SPECIFICATIONS

- 15 DEVICES MAXIMUM
- 31 PRIMARY ADDRESSES
- LENGTH OF CABLE - 2 METERS/DEVICE, 20 METERS TOTAL
- INTERCONNECT - STAR, LINEAR, OR COMBINATION
- MAXIMUM TRANSFER RATE - 1 MEGABYTE/SECOND
- ONLY 1 ACTIVE CONTROLLER AT A TIME, CONTROL MAY BE PASSED
- LOGIC - TTL, SCHOTTKY COMPATIBLE - O.C. & THREE STATE
- DEVICES CONSIST OF:
  - LISTENERS
  - TALKERS
  - TALKER/LISTENER
  - CONTROLLER
- ONLY ONE DEVICE MAY TALK AT A TIME, MANY MAY LISTEN
- BUS OPERATES ON NEGATIVE LOGIC: A LOW POSITIVE VOLTAGE IS TRUE (BINARY 1) AND A HIGH POSITIVE VOLTAGE IS FALSE (BINARY 0)
- 16 SIGNAL LINES - 8 FOR DATA, 8 FOR CONTROL AND STATUS MESSAGES, 8 GROUND
- MESSAGE TRANSFER - BYTE SERIAL/BIT PARALLEL, USING THREE WIRE HANDSHAKE
- 10 INTERFACE FUNCTIONS - 5 COMMUNICATION, 5 SPECIAL FUNCTION

NOTES:

IEEE 488 FUNCTIONAL ASPECTS

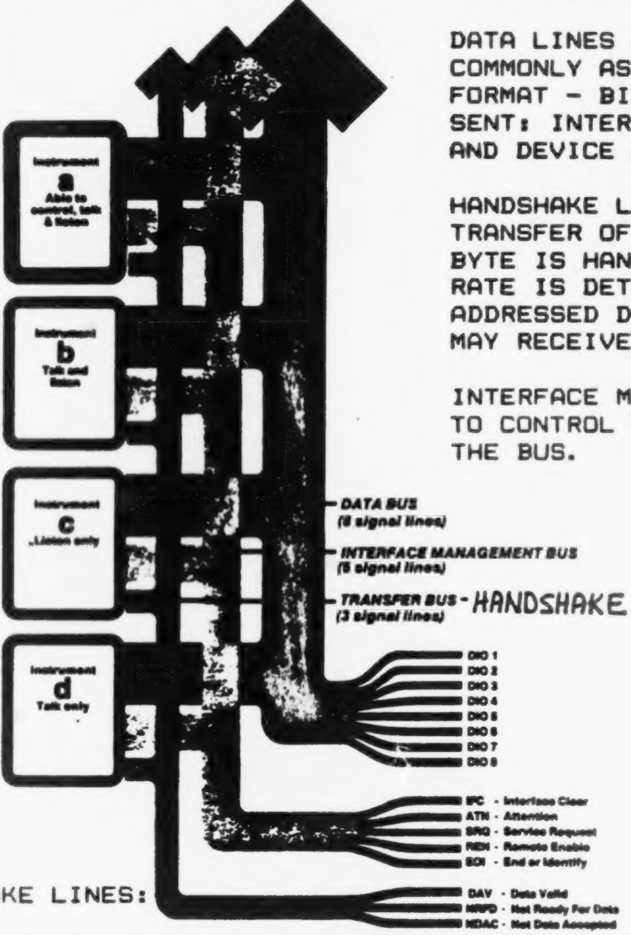
BUS LINES

REPRINTED WITH PERMISSION, COURTESY OF TEKTRONIX INC.

DATA LINES - BIDIRECTIONAL, COMMONLY ASCII, BUT MAY BE OTHER FORMAT - BINARY, BCD, ETC. ITEMS SENT: INTERFACE COMMANDS, ADDRESSES AND DEVICE DEPENDENT DATA.

HANDSHAKE LINES - USED TO CONTROL TRANSFER OF ASYNCHRONOUS DATA. EACH BYTE IS HANDSHAKED! DATA TRANSFER RATE IS DETERMINED BY SLOWEST ADDRESSED DEVICE. MULTIPLE DEVICES MAY RECEIVE DATA AT ONCE.

INTERFACE MANAGEMENT LINES - USED TO CONTROL FLOW OF INFORMATION ON THE BUS.


Instrument
a
Able to control, talk & listen
Instrument
b
Talk and listen
Instrument
c
Listen only
Instrument
d
Talk only
DATA BUS
(8 signal lines)
INTERFACE MANAGEMENT BUS
(5 signal lines)
TRANSFER BUS - HANDSHAKE
(3 signal lines)
DIO 1
DIO 2
DIO 3
DIO 4
DIO 5
DIO 6
DIO 7
DIO 8
IFC - Interface Clear
ATN - Attention
SRQ - Service Request
REN - Remote Enable
EOI - End or Identify
DAV - Data Valid
NRFD - Not Ready For Data
NDAC - Not Data Accepted

HANDSHAKE LINES:

DAV (DATA VALID) - SET ACTIVE OR TRUE (LOW) BY SOURCE WHEN DATA HAS BEEN PLACED ON THE DATA BUS (DIO1-8) AND NRFD IS FALSE (HIGH).

NRFD (NOT READY FOR DATA) - WHEN SET ACTIVE OR TRUE (LOW) IT INDICATES NOT ALL DEVICES ON THE BUS ARE READY FOR DATA. MUST WAIT FOR SLOWEST DEVICE TO BE SET FALSE (HIGH). SET BY ACCEPTORS OF DATA. OPEN COLLECTOR.

NDAC (NOT DATA ACCEPTED) - IT IS SET ACTIVE OR TRUE (LOW) BY ACCEPTORS UNTIL THEY RECEICE THE DATA. AS DATA IS RECEIVED THE CORRESPONDING ACCEPTOR RELEASES THE LINE, THE LINE GOES FALSE (HIGH) WHEN THE LAST DEVICE RELEASES IT. THUS THE RATE OF DATA TRANSFER IS DEPENDENT ON THE SLOWEST DEVICE INVOLVED IN THE HANDSHAKE.

NOTES:

IEEE 488 FUNCTIONAL ASPECTS

BUS LINES

BUS MANAGEMENT LINES (UNILINE COMMANDS)

ATN (ATTENTION) - IT IS ONLY ASSERTED (SET) BY THE CONTROLLER. IT IS USED TO GET THE "ATTENTION" OF THE DEVICES ON THE BUS. WHEN SET (LOW) THE INFORMATION ON THE BUS WILL BE A BUS COMMAND OR AN ADDRESS. WHEN THE ATN LINE IS RETURNED HIGH, DATA IS EXCHANGED BETWEEN THE ADDRESSED TALKER AND LISTENER(S).

* ATN LOW - COMMAND MODE
* ATN HIGH - DATA MODE

IFC (INTERFACE CLEAR) - SENT BY THE CONTROLLER, IT WILL SET INSTRUMENTS TO THEIR DEFAULT STATE (NO BUS ACTIVITY). ACTIVE LOW - TRUE.

REN (REMOTE ENABLE) - SENT BY THE CONTROLLER TO GET DEVICES TO RESPOND TO PROGRAM CONTROL WHEN ADDRESSED TO LISTEN. IF FALSE (HIGH) DEVICES ON THE BUS GO TO LOCAL (FRONT PANEL) CONTROL.

SRQ (SERVICE REQUEST) - SENT BY A DEVICE RESIDING ON THE BUS TO INTERRUPT THE CONTROLLER. INDICATES DEVICE IS REQUESTING SERVICE.

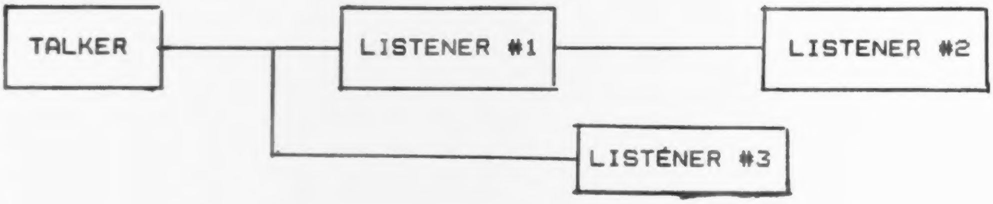
EOI (END OR IDENTIFY) - SENT BY TALKER TO LISTENER IT INDICATES THAT THE DATA BYTE ON THE DATA BUS (DIO1-8) IS THE LAST ONE BEING SENT IN THIS DATA MESSAGE. THIS INDICATES MESSAGE TERMINATION WITH OR IN LEIU OF LINE FEED MESSAGE TERMINATOR. MAY ALSO BE SENT BY THE CONTROLLER WITH ATN LINE TRUE TO PARALLEL POLL DEVICES FOR THEIR STATUS BIT.

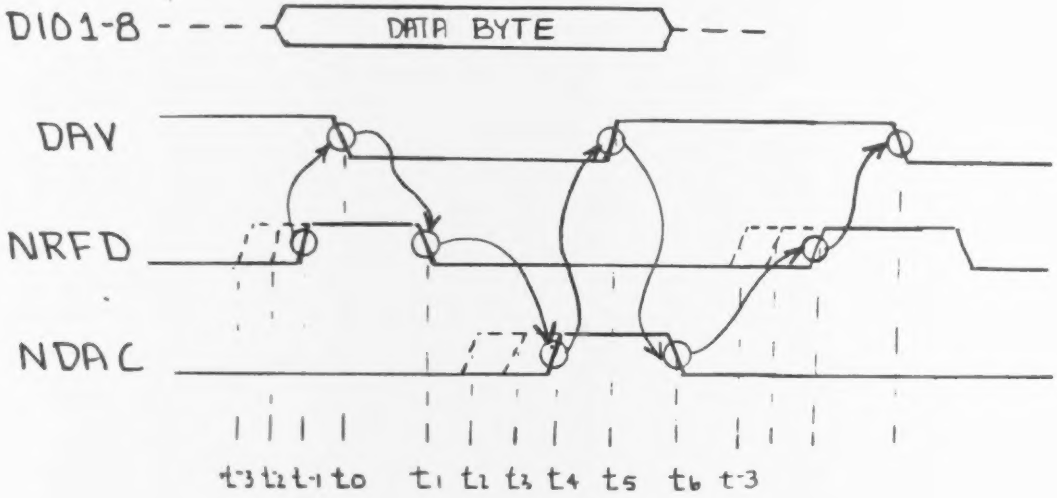
THE OPERATION OF THESE LINES IS INCORPORATED INTO THE OPERATING SOFTWARE OF GPIB INSTRUMENTS, AND THEREFORE IS, FOR THE MOST PART, TRANSPARENT TO THE USER. IT BECOMES IMPORTANT FOR THE USER TO UNDERSTAND THE OPERATION OF THESE LINES WHEN BUS OPERATIONS DO NOT FUNCTION AS THEY OUGHT TO. EXAMPLE: SUPPOSE A DMM SENDS A Cr & EOI THEN A Lf AS A TERMINATING SEQUENCE. IF THE CONTROLLER IS ONLY LOOKING FOR AN EOI, THEN IT WILL TERMINATE THE TRANSACTION FOLLOWING THE Cr. THE Lf STILL REMAINS IN THE DMM BUFFER. ON THE NEXT TRANSACTION THE DMM SENDS THE Lf FIRST. THE CONTROLLER, EXPECTING A NUMERIC VALUE, DOES NOT KNOW HOW TO RESPOND. TERMINATORS ARE HARDWARE OR SOFTWARE SELECTABLE ON MANY INSTRUMENTS.

NOTES:

IEEE 488 FUNCTIONAL ASPECTS

HANDSHAKE SEQUENCE


TALKER
LISTENER #1
LISTENER #2
LISTENER #3


DIO1-8
DATA BYTE
DAV
NRFD
NDAC
t-3 t-2 t-1 t0
t1 t2 t3 t4 t5 t6 t-3

t-3 LISTENER #1 READY FOR DATA

t-2 LISTENER #2 READY FOR DATA

t-1 LISTENER #3 READY FOR DATA, THE LAST DEVICE READY FOR DATA. THE TALKER SEEING NRFD FALSE PLACES DATA BYTE ON THE BUS.

t0 AFTER DATA HAS SETTLED, TALKER PULLS DAV TRUE INDICATING TO ALL LISTENERS THAT DATA IS VALID.

t1 FIRST LISTENER SEEING DAV TRUE PULLS NRFD TRUE TO INDICATE NOT READY FOR A NEW DATA BYTE (HAVING SENSED ONE IS AVAILABLE - DAV TRUE). ALL DEVICES ARE NOT READY FOR DATA!

t2 LISTENER #2 HAS READ THE DATA SO IT RELEASES NDAC. NDAC WILL NOT GO FALSE UNTIL RELEASED BY ALL DEVICES. #2 HAS ACCEPTED THE DATA.

t3 LISTENER #3 HAS READ THE DATA SO IT RELEASES NDAC.

NOTES:

HANDSHAKE SEQUENCE

t4 LISTENER #1 HAS READ THE DATA SO IT RELEASES THE NDAC LINE (FALSE) INDICATING IT HAS ACCEPTED THE DATA. SINCE IT IS THE LAST DEVICE TO ACCEPT THE DATA THE NDAC GOES HIGH (FALSE).

t5 THE TALKER SEEING THE NDAC FALSE SETS DAV FALSE, INDICATING DATA IS NO LONGER VALID.

t6 THE FIRST LISTENER SEEING DAV FALSE PULLS NDAC TRUE (LOW) TO INDICATE ACKNOWLEDGEMENT THAT DATA IS NO LONGER VALID.

CYCLE IS REPEATED FOR THE NEXT AND EACH DATA BYTE TRANSFERRED ACROSS THE BUS. THE SPEED OF THE HANDSHAKE SEQUENCE IS DETERMINED BY THE SLOWEST DEVICE ON THE BUS - t-1 & t4.

THE THREE WIRE HANDSHAKE TECHNIQUE IS PATENTED BY AND RIGHTS OF USE ARE SOLD BY HEWLETT PACKARD CO.

NOTES:

IEEE 488 FUNCTIONAL ASPECTS

TYPES OF INFORMATION SENT OVER THE BUS

BUS COMMAND - AN GROUP OF ASCII CHARACTERS WHICH HAVE EACH BEEN GIVEN A DESIGNATED FUNCTION. THEY WILL CAUSE DEVICES DESIGNED TO RESPOND TO THEM TO PERFORM A PREDEFINED OPERATION. THESE CHARACTERS ARE SENT WITH THE ATN LINE TRUE (LOW - COMMAND MODE).

ADDRESSED TO TALK - ONE OF SEVERAL ASCII CHARACTERS WHEN SENT WILL CAUSE ONE DEVICE (SWITCH SELECTED TO RESPOND TO IT) TO TALK. THESE ARE SENT WITH THE ATN LINE TRUE (LOW - COMMAND MODE). ONLY ONE TALKER MAY BE ACTIVE AT A TIME.

ADDRESSED TO LISTEN - ONE OF SEVERAL ASCII CHARACTERS WHEN SENT WILL CAUSE ONE DEVICE (SWITCH SELECTED TO RESPOND TO IT) TO LISTEN. THESE ARE SENT WITH THE ATN LINE TRUE (LOW - COMMAND MODE) TO LISTEN. THE LISTENING DEVICE(S) WILL RECEIVE DATA FROM THE TALKER.

DATA MESSAGE - INFORMATION SENT BETWEEN TALKER AND LISTENER(S). THIS MAY BE INSTRUMENT SETUP INFORMATION, READINGS, ETC. THESE ARE SENT WITH THE ATN LINE FALSE (HIGH - DATA MODE). DATA TRANSFER RATE IS DETERMINED BY THE SLOWEST LISTENER.

ALL DEVICES ACTIVE (CONNECTED TO THE BUS AND POWERED ON) WILL LISTEN TO COMMANDS FROM THE CONTROLLER. ONLY THOSE ADDRESSED TO LISTEN WILL RECEIVE DATA SENT BY THE TALKER.

TERMINATORS - PART OF THE DATA MESSAGE, SENT AT THE END OF THE MESSAGE TO INDICATE ITS TERMINATION (Cr/Lf, EOI).

NOTES:

IEEE 488 FUNCTIONAL ASPECTS

COMMAND MODE (ATN TRUE)

UNIVERSAL COMMANDS (UNADDRESSED)

COMMANDS SENT TO ALL DEVICES ACTIVE ON THE BUS. THESE ARE CLASSIFIED AS UNADDRESSED COMMANDS SINCE NO ADDRESS ACCOMPANIES THEM.

UNILINE COMMANDS - DISCUSSED ON Pg. 8-3. i.e. ATN, REN, ETC.

MULTILINE COMMANDS - THESE COMMANDS ARE SENT OVER THE DATA BUS (DIO1-8).

COMMAND	MNEMONIC	ASCII CHARACTER	HEX CODE
DEVICE CLEAR	DCL	DC4	14
LOCAL LOCKOUT	LLO	DC1	11
SERIAL POLL ENABLE	SPE	CAN	18
SERIAL POLL DISABLE	SPD	EM	19
PARALLEL POLL UNCONFIGURE	PPU	NAK	15
* UNTALK	UNT	_	5F
* UNLISTEN	UNL	?	3F

* CLASSIFIED AS ADDRESSES RATHER THAN COMMANDS EVENTHOUGH THEY PERFORM A COMMAND FUNCTION.

DCL - WILL RETURN ALL ACTIVE DEVICES, WHICH RECOGNIZE THE COMMAND, TO THEIR DEFAULT SETTINGS.

LLO - ON DEVICES WHICH RECOGNIZE THE COMMAND, WILL DISABLE INSTRUMENTS RETURN TO LOCAL BUTTON. THIS WILL PREVENT FRONT PANEL MANUAL OPERATION.

SPE - ON DEVICES WHICH RECOGNIZE THE COMMAND, THIS WILL SETUP THE SERIAL POLL MODE. THE DEVICES WILL OUTPUT THEIR STATUS BYTE WHEN TOLD TO TALK - MUST HAVE TALKER CAPABILITY.

SPD - WILL TERMINATE SERIAL POLL MODE FOR ALL RESPONDING DEVICES. THE DEVICES WILL RETURN DEVICE DEPENDENT DATA RATHER THAN STATUS INFORMATION WHEN ASKED TO TALK.

PPU - RESETS ALL DEVICES, CAPABLE OF RESPONDING, TO AN IDLE STATE, UNABLE TO RESPOND TO PARALLEL POLL.

UNT - THIS UNADDRESSES THE CURRENT TALKER. ADDRESSING ANOTHER TALKER OR AN UNUSED TALK ADDRESS ACCOMPLISHES THE SAME RESULT.

UNL - THIS UNLISTENS ALL ACTIVE LISTENERS. THEN THE DESIRED LISTENER(S) CAN BE ADDRESSED.

NOTES:

IEEE 488 FUNCTIONAL ASPECTS

ADDRESSED COMMANDS (ATN TRUE)

THESE COMMANDS ARE SENT TO DEVICES CURRENTLY ADDRESSED TO LISTEN.

COMMAND	MNEMONIC	ASCII CHARACTER	HEX CODE
GROUP EXECUTIVE TRIGGER	GET	BS	08
SELECTED DEVICE CLEAR	SDC	EOT	04
GO TO LOCAL	GTL	SQH	01
PARALLEL POLL CONFIGURE	PPC	ENQ	05
TAKE CONTROL	TCT	HT	09

GET - DEVICES WHICH RESPOND TO GET AND ARE ADDRESSED TO LISTEN, PERFORM A PREPROGRAMMED ACTION (e.g. TRIGGER, SWEEP, ECT.).

SDC - DEVICES WHICH RESPOND TO SDC AND ARE ADDRESSED TO LISTEN, GO TO THEIR DEFAULT SETTINGS.

GTL - DEVICES WHICH RESPOND TO GTL AND ARE ADDRESSED TO LISTEN, LEAVE THE REMOTE STATE AND RETURN TO FRONT PANEL (LOCAL) CONTROL. ONCE ADDRESSED TO LISTEN THEY WILL RETURN TO REMOTE.

PPC - IF DESIGNED TO RESPOND TO PPC AND ADDRESSED TO LISTEN, THE DEVICE WILL BE CONFIGURED TO THE PARALLEL POLL ENABLE (PPE) COMMAND WHICH FOLLOWS THE PPC.

TCT - IF DESIGNED TO RESPOND TO TCT THE TALKER (A PASSIVE CONTROLLER ONLY) MAY ASSUME THE ACTIVE CONTROLLER FUNCTION.

NOTES:

IEEE 488 FUNCTIONAL ASPECTS

SECONDARY COMMANDS

THESE ACCOMPANY THE PPC COMMAND

SECONDARY COMMAND	MNEMONIC	ASCII CHARACTER	HEX CODE
PARALLEL POLL ENABLE	PPE	BLANK THRU 0	60 THRU 6F
PARALLEL POLL DISABLE	PPD	P	50

PARALLEL POLL - THIS IS INITIATED BY THE CONTROLLER TO OBTAIN STATUS INFORMATION FROM THE DEVICES ON THE BUS. EACH DEVICE IS ASSIGNED ONE DATA LINE (DIO1-8), THUS ONLY 16 DEVICES MAY BE INVOLVED IN A PARALLEL POLL. IT IS NOT USED OFTEN IN INSTRUMENT SYSTEMS FOR THE ABOVE REASON AS WELL AS A SERIAL POLL MUST BE PERFORMED ANYWAY TO GET DETAILED STATUS INFORMATION. THE PARALLEL POLL ONLY INDICATES WHICH DEVICE REQUESTED SERVICE. A SERIAL POLL DELIVERS THE DETAILED STATUS INFORMATION.

THE PPC COMMAND IS SENT TO INDICATE A PARALLEL CONFIGURE IS TO OCCUR. ONE OF 16 PPE SECONDARY COMMANDS (MSA 0-15) IS SENT TO PROVIDE THE DATA LINE ASSIGNMENT (2 DEVICES MAY BE ASSIGNED TO ONE DATA LINE).

A PARALLEL POLL IS CONDUCTED BY THE CONTROLLER BY ASSERTING TRUE (LOW) BOTH ATN AND EOI. CONFIGURED DEVICES MUST RESPOND WITHIN 200nS BY ASSERTING THEIR ASSIGNED DATA LINE. THE CONTROLLER CAN READ THE RESULTS OF THE POLL 2 uS LATER.

IF SERIAL POLL WERE USED ALONE, EACH DEVICE ON THE BUS WOULD HAVE TO BE POLLED (UP TO 15) TO DETERMINE WHO REQUESTED THE SRQ. WITH PARALLEL POLL IT CAN BE NARROWED DOWN TO 2 (UP TO 2 DEVICES PER DATA LINE) OR 1 DEVICE, THEN THESE CAN BE SERIAL POLLED.

NOTES:

ASCII/IEEE CODES - REPRINTED WITH PERMISSION, JOHN FLUKE MFG. CO. INC.

Table 3-4. ASCII/IEEE Std 488-1978 Bus Codes

ASCII CHAR.	DECIMAL	OCTAL	HEX	BINARY 7654 3210	DEV. NO.	MESSAGE ATN=TRUE	
NUL	0	000	00	0000 0000			ADDRESSED COMMANDS
SOH	1	001	01	0000 0001		GTL	
STX	2	002	02	0000 0010			
ETX	3	003	03	0000 0011			
EOT	4	004	04	0000 0100		SDC	
ENQ	5	005	05	0000 0101		PPC	
ACK	6	006	06	0000 0110			
BELL	7	007	07	0000 0111			
BS	8	010	08	0000 1000		GET	
HT	9	011	09	0000 1001		TCT	
LF	10	012	0A	0000 1010			
VT	11	013	0B	0000 1011			
FF	12	014	0C	0000 1100			
CR	13	015	0D	0000 1101			
SO	14	016	0E	0000 1110			
SI	15	017	0F	0000 1111			
DLE	16	020	10	0001 0000			UNIVERSAL COMMANDS
DC1	17	021	11	0001 0001		LLO	
DC2	18	022	12	0001 0010			
DC3	19	023	13	0001 0011			
DC4	20	024	14	0001 0100		DCL	
NAK	21	025	15	0001 0101		PPU	
SYN	22	026	16	0001 0110			
ETB	23	027	17	0001 0111			
CAN	24	030	18	0001 1000		SPE	
EM	25	031	19	0001 1001		SPD	
SUB	26	032	1A	0001 1010			
ESC	27	033	1B	0001 1011			
FS	28	034	1C	0001 1100			
GS	29	035	1D	0001 1101			
RS	30	036	1E	0001 1110			
US	31	037	1F	0001 1111			
SPACE	32	040	20	0010 0000	0	MLA	LISTEN ADDRESSES
!	33	041	21	0010 0001	1	MLA	
"	34	042	22	0010 0010	2	MLA	
#	35	043	23	0010 0011	3	MLA	
$	36	044	24	0010 0100	4	MLA	
%	37	045	25	0010 0101	5	MLA	
&	38	046	26	0010 0110	6	MLA	
'	39	047	27	0010 0111	7	MLA	
(	40	050	28	0010 1000	8	MLA	
)	41	051	29	0010 1001	9	MLA	
*	42	052	2A	0010 1010	10	MLA	
+	43	053	2B	0010 1011	11	MLA	
,	44	054	2C	0010 1100	12	MLA	
-	45	055	2D	0010 1101	13	MLA	
.	46	056	2E	0010 1110	14	MLA	
/	47	057	2F	0010 1111	15	MLA	
0	48	060	30	0011 0000	16	MLA	
1	49	061	31	0011 0001	17	MLA	
2	50	062	32	0011 0010	18	MLA	
3	51	063	33	0011 0011	19	MLA	
4	52	064	34	0011 0100	20	MLA	
5	53	065	35	0011 0101	21	MLA	
6	54	066	36	0011 0110	22	MLA	
7	55	067	37	0011 0111	23	MLA	
8	56	070	38	0011 1000	24	MLA	
9	57	071	39	0011 1001	25	MLA	
:	58	072	3A	0011 1010	26	MLA	
;	59	073	3B	0011 1011	27	MLA	
<	60	074	3C	0011 1100	28	MLA	
=	61	075	3D	0011 1101	29	MLA	
>	62	076	3E	0011 1110	30	MLA	
?	63	077	3F	0011 1111		UNL	

ASCII CHAR.	DECIMAL	OCTAL	HEX	BINARY 7654 3210	DEV. NO.	MESSAGE ATN=TRUE	
@	64	100	40	0100 0000	0	MTA	TALK ADDRESSES
A	65	101	41	0100 0001	1	MTA	
B	66	102	42	0100 0010	2	MTA	
C	67	103	43	0100 0011	3	MTA	
D	68	104	44	0100 0100	4	MTA	
E	69	105	45	0100 0101	5	MTA	
F	70	106	46	0100 0110	6	MTA	
G	71	107	47	0100 0111	7	MTA	
H	72	110	48	0100 1000	8	MTA	
I	73	111	49	0100 1001	9	MTA	
J	74	112	4A	0100 1010	10	MTA	
K	75	113	4B	0100 1011	11	MTA	
L	76	114	4C	0100 1100	12	MTA	
M	77	115	4D	0100 1101	13	MTA	
N	78	116	4E	0100 1110	14	MTA	
O	79	117	4F	0100 1111	15	MTA	
P	80	120	50	0101 0000	16	MTA	
Q	81	121	51	0101 0001	17	MTA	
R	82	122	52	0101 0010	18	MTA	
S	83	123	53	0101 0011	19	MTA	
T	84	124	54	0101 0100	20	MTA	
U	85	125	55	0101 0101	21	MTA	
V	86	126	56	0101 0110	22	MTA	
W	87	127	57	0101 0111	23	MTA	
X	88	130	58	0101 1000	24	MTA	
Y	89	131	59	0101 1001	25	MTA	
Z	90	132	5A	0101 1010	26	MTA	
[	91	133	5B	0101 1011	27	MTA	
\	92	134	5C	0101 1100	28	MTA	
]	93	135	5D	0101 1101	29	MTA	
^	94	136	5E	0101 1110	30	MTA	
_	95	137	5F	0101 1111		UNT	
	96	140	60	0110 0000	0	MSA	SECONDARY ADDRESSES
a	97	141	61	0110 0001	1	MSA	
b	98	142	62	0110 0010	2	MSA	
c	99	143	63	0110 0011	3	MSA	
d	100	144	64	0110 0100	4	MSA	
e	101	145	65	0110 0101	5	MSA	
f	102	146	66	0110 0110	6	MSA	
g	103	147	67	0110 0111	7	MSA	
h	104	150	68	0110 1000	8	MSA	
i	105	151	69	0110 1001	9	MSA	
j	106	152	6A	0110 1010	10	MSA	
k	107	153	6B	0110 1011	11	MSA	
l	108	154	6C	0110 1100	12	MSA	
m	109	155	6D	0110 1101	13	MSA	
n	110	156	6E	0110 1110	14	MSA	
o	111	157	6F	0110 1111	15	MSA	
p	112	160	70	0111 0000	16	MSA	
q	113	161	71	0111 0001	17	MSA	
r	114	162	72	0111 0010	18	MSA	
s	115	163	73	0111 0011	19	MSA	
t	116	164	74	0111 0100	20	MSA	
u	117	165	75	0111 0101	21	MSA	
v	118	166	76	0111 0110	22	MSA	
w	119	167	77	0111 0111	23	MSA	
x	120	170	78	0111 1000	24	MSA	
y	121	171	79	0111 1001	25	MSA	
z	122	172	7A	0111 1010	26	MSA	
(	123	173	7B	0111 1011	27	MSA	
|	124	174	7C	0111 1100	28	MSA	
)	125	175	7D	0111 1101	29	MSA	
~	126	176	7E	0111 1110	30	MSA	
	127	177	7F	0111 1111			

ANATOMY OF HIGHER LEVEL COMMANDS

CONTROLLERS HAVE SPECIAL HIGH LEVEL COMMANDS WHICH ARE DESIGNED FOR GPIB CONTROL OPERATIONS. THESE ARE CLASSIFIED AS ENHANCEMENTS TO THE LANGUAGE FOR WHICH THEY ARE INCLUDED IN i.e. BASIC. THESE COMMANDS CONTAIN THE GPIB COMMANDS AND OPERATIONS IMBEDDED WITHIN THE HIGH LEVEL COMMAND. IT BECOMES NECESSARY TO UNDERSTAND THE INNERWORKINGS OF HIGH LEVEL COMMANDS WHEN ANALYZING BUS ACTIVITY WITH A BUS ANALYZER.

THE FOLLOWING IS A LIST OF THE INNERWORKINGS OF HEWLETT PACKARD BASIC COMMAND. REFERENCE WILL BE MADE TO THE ASCII/IEEE std 488-1978 BUS CODE CHART ON Pg.9-1.

ENTER COMMAND: ENTER 723; X
ASSUME THE VALUE ENTERED INTO X IS +3.57960E-2CrLf

ATN TRUE=*	MESSAGE	ASCII	HEX	DESCRIPTION
*	UNL	?	3F	UNLISTENS ALL DEVICES PREVIOUSLY ADDRESSED TO LISTEN.
*	MLA 21	5	35	COMPUTER IS DESIGNATED AS A LISTENER. ITS ADDRESS IS 21. THE ACTUAL ADDRESS IS 32+21=53, WHICH IS 35 IN HEX i.e. ALL LISTEN ADDRESSES ARE 32 DECIMAL PLUS THE SPECIFIED BUS ADDRESS.
*	MTA 23	W	57	DEVICE 723 IS DESIGNATED AS A TALKER. ITS ADDRESS IS 23. THE ACTUAL ADDRESS IS 64+23=87, WHICH IS 57 IN HEX i.e. ALL TALK ADDRESSES ARE 64 DECIMAL PLUS THE SPECIFIED BUS ADDRESS.
	DATA	+	2B	HANDSHAKING OF DATA BEGINS, ATN FALSE, + SIGN SENT.
	DATA	3	33	DATA VALUE 3 IS SENT.
	DATA	.	2E	DATA VALUE . IS SENT.
	DATA	5	35	DATA VALUE 5 IS SENT.
	DATA	7	37	DATA VALUE 7 IS SENT.
	DATA	ETCETERA		
	DATA	Cr	0D	CARRAGE RETURN PORTION OF TERMINATION SEQUENCE IS SENT.
	DATA	Lf	0A	LINE FEED TERMINATOR IS SENT.

NOTES:

IEEE 488 FUNCTIONAL ASPECTS

ANATOMY OF HIGHER LEVEL COMMANDS

THE FOLLOWING IS A LIST OF THE INNERWORKINGS OF HEWLETT PACKARD BASIC COMMAND. REFERENCE WILL BE MADE TO THE ASCII/IEEE std 488-1978 BUS CODE CHART ON Pg.9-1.

OUTPUT COMMAND: OUTPUT 723; H1

ATN TRUE=*	MESSAGE	ASCII	HEX	DESCRIPTION
*	UNL	?	3F	UNLISTENS ALL DEVICES PREVIOUSLY ADDRESSED TO LISTEN.
*	MTA 21	U	55	COMPUTER IS DESIGNATED AS A TALKER. ITS ADDRESS IS 21. THE ACTUAL ADDRESS IS 64+21=85, WHICH IS 55 IN HEX i.e. ALL TALK ADDRESSES ARE 64 DECIMAL PLUS THE SPECIFIED BUS ADDRESS.
*	MLA 23	7	37	DEVICE 723 IS DESIGNATED AS A LISTENER. ITS ADDRESS IS 23. THE ACTUAL ADDRESS IS 32+23=55, WHICH IS 37 IN HEX i.e. ALL LISTEN ADDRESSES ARE 32 DECIMAL PLUS THE SPECIFIED BUS ADDRESS.
	DATA	H	48	HANDSHAKING OF DATA BEGINS, ATN FALSE, H CHARACTER SENT.
	DATA	1	31	DATA VALUE 1 IS SENT.
	DATA		20	DATA VALUE FOR A BLANK IS SENT.
	DATA		20	18 MORE BLANKS ARE SENT.
	DATA	Cr	0D	CARRIAGE RETURN PORTION OF TERMINATION SEQUENCE IS SENT.
	DATA	Lf	0A	LINE FEED TERMINATOR IS SENT.

THE H1 COMMAND IS USED BY THE HEWLETT PACKARD 3478A DMM TO SET IT UP INTO DC VOLTS, AUTORANGE, SINGLE TRIGGER, ETC. IT IS SENT IN PLACE OF SEVERAL INDIVIDUAL SETUP COMMANDS. THE STANDARD OUTPUT STATEMENT USES A FREE FIELD FORMAT i.e. 21 SPACES. EACH ONE OF THESE UNUSED SPACES MUST BE HANDSHAKED. TO ELIMINATE THIS WASTED TIME UTILIZE THE OUTPUT USING STATEMENT.

NOTES:

IEEE 488 FUNCTIONAL ASPECTS

THE USE OF A BUS ANALYZER

PRELUDE - THE GPIB IS A VERY POWERFUL INTERFACE MECHANISM, WHICH IS EASILY PROGRAMMED IN ONE OF SEVERAL HIGHER LEVEL LANGUAGES (BASIC BEING THE MOST COMMON). THE FOLLOWING CONCEPTS WERE EXTRACTED FROM THE "RACAL-DANA GPIB TEST SYSTEMS HANDBOOK", "OPERATORS MANUAL MODEL 488 GPIB ANALYZER" RACAL-DANA INSTRUMENTS, INC. THERE ARE THREE GENERAL TYPES OF GPIB FAILURE MODES: 1. HARDWARE MALFUNCTIONS, PROGRAM CODING ERRORS, DATA FORMAT INCOMPATABILITY. OF THESE, THE LAST TWO ARE THE MOST COMMON. A BUS ANALYZER IS A TOOL WHICH CAN MONITOR AND RECORD BUS ACTIVITY i.e. DATA/COMMAND BYTES AND COMMAND LINES. IT CAN BE USED TO SEE EXACTLY WHAT IS BEING SENT OVER THE BUS. THIS INFORMATION CAN THEN BE COMPARED TO WHAT SHOULD BE SENT OVER THE BUS AND NECESSARY SOFTWARE CORRECTIONS CAN BE MADE.

THERE ARE SEVERAL LOW COST BUS ANALYZERS ON THE MARKET. THE MODEL 488 FROM RACAL-DANA WAS CHOSEN DUE TO ITS LOW COST AND EASE OF USE. THIS UNIT UTILIZES SWITCH DEFINED BREAKPOINTS (WORD RECOGNITION) TO CAPTURE BUS EVENTS TO STORE INTO ITS 40 BYTE MEMORY. DATA IS READOUT IN HEXADECIMAL AND CAN THEN BE CONVERTED TO A COMMAND OR ASCII DATA UTILIZING AN ASCII-GPIB CHART (Pg.9-1).

OPERATING MODES: (SEE DIAGRAM NEXT PAGE)

TRACE - DATA IS CAPTURED AND STORED IN MEMORY FOR LATER ANALYSIS.

PASSIVE - BUS TRANSACTIONS ARE SHOWN IN REAL TIME ON THE 488's LCD DISPLAY.

SINGLE STEP (S/S) - ALLOWS BUS TRANSACTIONS TO BE STOPPED OR SINGLE STEPPED.

BASIC OPERATION:

TRIGGER SETTING - THE TRIGGER CONDITION IS SET VIA EIGHT SLIDE SWITCHES. THESE ARE SET TO LOGIC (NEGATIVE) 1,0, OR X (DON'T CARE). THE TRIGGER EVENT MAY PRECEED THE GATHERING OF DATA BY SELECTING "START". IT MAY END THE GATHERING OF DATA BY SELECTING "END OR BE CENTERED IN THE DATA BY PLACING THE MEMORY START/END SWITCH IN ITS MIDPOSITION.

GATHERING DATA - PLACE THE TRACE-S/S SWITCH TO TRACE. PUSH THE ARM SWITCH TO ARM (MOMENTARY SWITCH), THIS SETS THE ANALYZER UP TO START LOOKING FOR THE TRIGGER CHARACTER. DATA IS SHIFTED THROUGH THE 40 BYTE MEMORY. WHEN THE TRIGGER CHARACTER IS FOUND THE ARM ANNUNCIATOR SHUTS OFF.

VIEWING DATA- THE MEMORY UP/DOWN KEY MAY BE USED TO LOOK FORWARD/ BACKWARD IN MEMORY.

NOTES:

1/3

BUS ANALYZER
DATA LOG SHEET

TEST ENGINEER E.HERRMANN DATE 5-86
ANALYSIS FROM LINE ______ TO ______
PROGRAM TITLE OUTPUT USING STATEMENT

HEX CODE	ATN *=TRUE	OTHER UNILINES	COMMAND OR DATA	PROGRAM STATEMENT	COMMENTS
55	✓		MTA 21		
32	✓		MLA 18		
56			V		
41			A		
52			R	OUTPUT 718	
32			2	USING "#,7A" ;A$	
53			S	A$=VAR2S5F	
35			5		
46			F		
3F	✓		UNL		
55	✓		MTA 21		
32	✓		MLA 18		
3F			?	OUTPUT 718;	
20			ƀ	"?"	20 SPACES
0D			Cr		
0A			Lf		
3F	✓		UNL		
35	✓		MLA 21		
52	✓		MTA 18		
2B			+		

NOTES:

2/3

BUS ANALYZER
DATA LOG SHEET

TEST ENGINEER ____________________ DATE ____________
ANALYSIS FROM LINE ______ TO ________
PROGRAM TITLE ____

HEX CODE	ATN *=TRUE	OTHER UNILINES	COMMAND OR DATA	PROGRAM STATEMENT	COMMENTS
30			0		
30			0		
2E			.		
30			0		
30			0		
34			4		
35			5		
38			8		
45			E		
2B			+		
30			0		
0D			Cr		
0A			Lf		
3F	✓		UNL		
55	✓		MTA21		
25	✓		MLA5		
20			b̸		21 SPACES
0D			[illegible]		
0A			[illegible]		

NOTES:

```
90 PRINT B$
100 END

+00.00472E+0

10 PRINTER IS 705
20 A$="VAR2S5F"
30 OUTPUT 718 USING "#,7A" ; A$
40 OUTPUT 718 ;"?"
50 ENTER 718 ; B$
60 PRINT ""
70 PLIST
80 PRINT ""
90 PRINT B$
100 END

+00.00458E+0
```

SYSTEM PERFORMANCE

PRELUDE - MANY FACTORS CONTRIBUTE TO THE PERFORMANCE OF A GPIB SYSTEM. IN MANY INSTANCES ADDRESSING ONLY ONE FACTOR WHICH IS AFFECTING SYSTEM PERFORMANCE WITHOUT CONSIDERING OTHERS, MAY NOT BOOST PERFORMANCE APPRECIABLY. AN EXAMPLE WOULD BE ATTEMPTING TO BOOST SYSTEM THROUGHPUT BY UTILIZING A DMM WITH A FASTER READING RATE, WITHOUT CONSIDERING CONTROLLER DATA TRANSFER RATE. LISTED BELOW ARE SOME MAJOR TOPICS WHICH WILL BE INVESTIGATED IN DETERMINING THEIR EFFECT ON SYSTEM PERFORMANCE.

1. SYSTEM CONTROLLER
   - A. DEDICATED CONTROLLER
   - B. PERSONNEL COMPUTER
   - C. NUMBER OF GPIB PORTS

2. INSTRUMENTATION
   - A. LEVEL OF PROGRAMMABILITY
   - B. INSTRUMENT INTELLIGENCE
   - C. INSTRUMENT FEATURES
   - D. TIME SENSITIVE CHARACTERISTICS

3. SOFTWARE
   - A. PROGRAM MODULARITY
   - B. COMMAND LANGUAGE LEVEL
   - C. TIMING

4. DOCUMENTATION
   - A. HARDWARE
     - 1. INSTRUMENTATION & CONTROLLER
     - 2. SYSTEM
   - B. SOFTWARE (FINAL DOCUMENTATION)
     - 1. PROJECT PROPOSAL
     - 2. DESIGN PLAN
     - 3. COMPLETION DEFINITION
     - 4. CUSTOMER REFERENCE SPECIFICATIONS
     - 5. STRUCTURE CHARTS
     - 6. MODULE DOCUMENTATION

5. GPIB PROBLEMS & SOLUTIONS
   - A. HARDWARE
     - 1. BUS EXTENDERS
     - 2. RS232C - GPIB CONVERTERS
     - 3. REMOTE LINKAGE VIA MODEM
     - 4. FAILURE TYPES
   - B. SOFTWARE
     - 1. PROGRAM CODING
     - 2. DATA FORMAT INCOMPATABILITY

NOTES:

SYSTEM PERFORMANCE

SYSTEM CONTROLLER

DEDICATED CONTROLLERS

PRELUDE - SEVERAL MANUFACTURERS MARKET COMPUTERS WHICH HAVE BEEN SPECIFICALLY DESIGNED AS INSTRUMENT CONTROLLERS. THE FOLLOWING COMPANIES MANUFACTURE DESK TOP CONTROLLERS: HEWLETT PACKARD CO., JOHN FLUKE MFG. CO., TEKTRONIX INC., AND OTHERS. MINICOMPUTER MANUFACTURERS SUCH AS DIGITAL EQUIPMENT CORP., DATA GENERAL CORP., HEWLETT PACKARD CO., AND OTHERS SUPPLY GPIB INTERFACES FOR MINIS.

ADVANTAGES OF DEDICATED CONTROLLERS

* SPECIALIZED HIGH LEVEL COMMANDS WHICH INCORPORATE SEVERAL GPIB LEVEL COMMANDS. EXAMPLE: HP's OUTPUT COMMAND AUTOMATICALLY SENDS THE FOLLOWING SEQUENCE - UNL, MTAxx, MLAyy, 21 BYTES DATA, Cr Lf.

* MULTIPLE GPIB PORTS
    FLUKE 1700 SERIES - 2 PORTS
    HP DESK TOPS - UP TO 4 PORTS
    TEKTRONIX 4041 - UP TO 2 PORTS
    MINICOMPUTERS - SOME MORE THAN 15 PORTS. THIS ALLOWS A GROUP OF TEST STATIONS TO BE LINKED TO THE SAME DATA BASE. THIS ALLOWS EASY OVERALL FAILURE ANALYSIS. MINIs ARE NEEDED FOR A MULTIPLE TEST STATION TIME SHARING SYSTEM FOR REALISTIC DATA THROUGHPUT.

* REAL TIME SERVICE REQUEST (SRQ) IS DESIGNED IN. THIS ALLOWS THE CONTROLLER TO RESPOND MORE RAPIDLY. THIS IS A HARDWARE DRIVEN INTERRUPT RATHER THAN A SOFTWARE POLLED SRQ.

* DEDICATED CONTROLLERS CAN BE FASTER THAN PERSONNAL COMPUTERS DUE TO: FASTER CLOCK SPEED, E-DISK (RAM CONFIGURED AS A DISK), BUBBLE MEMORY, HARD DISK.

DISADVANTAGES OF DEDICATED CONTROLLEERS

* COST - DUE TO ENHANCED CAPABILITIES FOR CONTROLLER FUNCTIONS THE DEDICATED CONTROLLER CAN COST SEVERAL TIMES THAT OF A PERSONAL COMPUTER WHICH CAN DO THE JOB. EVENTHOUGH SLOWER, WITH FEWER FEATURES THE PERSONEL COMPUTER MAY BE ALL THAT IS NEEDED FOR THE APPLICATION.

* LIMITED ANOUNT OF NON CONTROLLER TYPE SOFTWARE i.e. WORD PROCESSORS. SPREAD SHEETS, ETC. THIS IS IMPORTANT IF THE UNIT MUST BE USEABLE FOR OTHER DUTIES OTHER THAN INSTRUMENT CONTROL.

* MAY BE ANOTHER OPERATING SYSTEM TO LEARN.

NOTES:

APPENDIX A

SOLUTIONS TO LAB EXERCISES

The following is a set of possible solutions to the lab exercises in this curriculum.

```
10 !  LAB 2-1 CONTROLLER FAMILIARIZATION
20 !
30 ! THE PURPOSE OF THIS LAB IS TO FAMILIARIZE THE STUDENT WITH CONTROLLER
     FEATURES, OPERATION, AND EDITING CAPABILITIES.
40 !
50 ! KEYBOARD FAMILIARIZATION
60 !    CALCULATOR MODE: PERFORM THE FOLLOWING MATHEMATICAL EXPRESSION
                         Xc=1/2 FC, F=123.27KHz,C=0.0047uF, Xc=[illegible].[illegible]75855
70 !      WRITE YOUR EXPRESSION AS ENTERED INTO THE COMPUTER:
            1/ 2* 123.27E3*.0047E-6)
80 !    PROGRAM MODE: DURING THIS PORTION OF THE LAB YOU WILL BE USING THE
          EDITING FEATURES TO MODIFY THE PROGRAM BELOW.
90 !      USE THE FOLLOWING KEYS TO EDIT THE PROGRAM TO THE LEFT SO IT
          MATCHES THE ONE TO THE RIGHT: BACK SPACE, -LINE, -CHAR, I/R,
100 !     AND CURSOR KEYS. REMOVE THE FAR LEFT ! AS EACH LINE IS CORRECTED
110 !     PRIOR TO END LINE, STARTING FROM LINE 180!!!!!!!!!!!!!!
120 !
130 !   THIS PROGRAM INSTRUCTS THE           130 !    THIS PROGRAM INSTRUCTS THE
140 !   DMM TO TAKE 100 READINGS,            140 !    DMM TO TAKE 100 READINGS,
150 !   STORE THEM IN ARRAY, AND             150 !    STORE THEM IN AN ARRAY, AND
160 !                                        160
170 !   AVERAGE THEM.                        170 !    AVERAGE THEM.
180 !                                        180 !
190 GOSUB SETUP  !                           190 GOSUB SETUP !
200 GOSUB KEYTEST  !                         200 GOSUB KEYTEST !
210 GOSUB AVG  !                             210 GOSUB AVG !
220 !                                        220 !
230 GOSUB FINAL_VAL  !                       230 GOSUB FINAL_VAL !
240 GOTO THE_END  !                          240 GOTO THE_END !
245 ! , 246 !, 247 !                        245 !, 246 !, 247 !
250 SETUP: ! DMM INITIALIZATION              250 SETUP: ! DMM INITIALIZATION
250 SETUP: ! DMM INITIALIZATION              250 SETUP: ! DMM INITIALIZATION
260 RESET 7 ! RESETS INTERFACE #7            260 RESET 7 ! RESETS INTERFACE #7
270 CLEAR 7 ! DEVICES TO DEFAULT             270 CLEAR 7 ! DEVICES TO DEFAULT
280 REMOTE 723 ! TO BUS CONTROL              280 REMOTE 723 ! TO BUS CONTROL
290 LOCAL LOCKOUT 7 ! FRONT DISABLE          290 LOCAL LOCKOUT 7 ! FRONT DISABLE
300 OUTPUT 723 ;"HO" ! DC, NO TRIG           300 OUTPUT 723;"HO" ! DC, NO TRIG
310 RETURN ! TO MAIN PRGM                    310 RETURN ! TO MAIN PRGM
315 ! , 316 !, 317 !                         315 !, 316 !, 317 !
320 KEYTEST: ! K5 TO START                   320 KEYTEST: ! K5 TO START
330 ON KEY# 5,"RUN TEST" GOTO START  !       330 ON KEY#5,"RUN TEST" GOTO START !
340 KEY LABEL ! DISPLAYS LABEL               340 KEY LABEL ! DISPLAYS LABEL
350 FLASH: DISP "PRESS K5 TO START" !        350 FLASH: DISP "PRESS K5 TO START" !
360 WAIT 3000 ! 3 SEC DELAY                  360 WAIT 3000 ! 3 SEC DELAY
370 CLEAR ! BLANKS SCREEN                    370 CLEAR ! BLANKS SCREEN
380 KEY LABEL ! DISPLAYS LABEL               380 KEY LABEL ! DISPLAYS LABEL
390 WAIT 100 ! DELAY .1 SEC                  390 WAIT 100 ! DELAY .1 SEC
400 GOTO FLASH  ! LOOP                       400 GOTO FLASH ! LOOP
410 START: DISP "TEST STARTED" ! MSG         410 START: DISP "TEST STARTED" ! MSG
420 RETURN ! TO MAIN PRGM                    420 RETURN ! TO MAIN PRGM
425 ! , 426 !, 427 !                         425 !, 426 !, 427 !
430 AVG: ! TAKE READINGS & AVG               430 AVG: ! TAKE READINGS & AVG
440 DIM VOLT(100) ! SETUP ARRAY              440 DIM VOLT(100) ! SETUP ARRAY
450 RD=0 ! SET VARIABLE TO 0                 450 RD=0  ! SET VARABLE TO 0
460 FOR N=1 TO 100 ! SETUP LOOP              460 FOR N=1 TO 100 ! SETUP LOOP
470 OUTPUT 723 ;"T3" ! TRIG DMM              470 OUTPUT 723; "T3" ! TRIG DMM
480 ENTER 723 ; VOLT(RD) ! INPUT             480 ENTER 723; VOLT(RD) ! INPUT
485 RD=RD+1 ! INCREMENT                      485 RD=RD+1 !INCREMENT
490 NEXT N ! LOOP BACK OR NOT                490 NEXT N ! LOOP BACK OR NOT
500 SUM=0 ! SET VARIABLE TO 0                500 SUM=0 ! SET VARIABLE TO 0
```

LS-2

```
510 FOR M=0 TO 99 ! SETUP LOOP
520 SUM=SUM+VOLT(M) ! ADD
530 NEXT M ! LOOP BACK OR NOT
535 SUM=SUM/M !
540 RETURN ! TO MAIN PRGM
550 FINAL_VAL: ! PRINT RESULT
560 DISP "AVG VALUE IS ";SUM !
570 PRINTER IS 705 ! SETUP PRINTER
580 PRINT "AVG VALUE IS ";SUM !
590 PRINT USING "/////" ! CrLf
600 RETURN ! TO MAIN PRGM
610 THE_END: !     PROGRAM END
620 LOCAL 7 ! ENABLE FRONT PANEL
630 END !        ALL DONE
```

```
510 FOR M=0 TO 99 ! SETUP LOOP
520 SUM=SUM+VOLT(M) ! ADD
530 NEXT M  ! LOOP BACK OR NOT
535 SUM=SUM/M
540 RETURN ! TO MAIN PRGM
550 FINAL_VAL: ! PRINT RESULT
560 DISP "AVG VALUE IS "; SUM  !
570 PRINTER IS 705 ! SETUP PRINTER
580 PRINT "AVG VALUE IS "; SUM  !
590 PRINT USING "/////" ! CrLf
600 RETURN ! TO MAIN PRGM
610 THE_END: ! PROGRAM END
620 LOCAL 7   ! ENABLE FRONT PANEL
630 END ! ALL DONE
```

LS-3

```
10 ! *************************
20 !
30 !  CAPACITANCE CALCULATING
40 !         PROGRAM
50 !
60 ! CCTS LAB EXERCISE SOLUTION
70 !      BY BRUCE PERLER
80 !
90 ! *************************
100 !
110 ! *************************
120 !
130 !        MENU ROUTINE
140 !
150 CLEAR
160 DISP " "
170 DISP " "
180 DISP "  K1...  INSTRUCTIONS"
190 DISP " "
200 DISP "  K2...  RUN TEST"
210 ON KEY# 1 GOTO 400
220 ON KEY# 2 GOTO 260
230 GOTO 210
240 !
250 ! ************************
260 !
270 ! ************************
280 !
290 !       TEST DRIVER
300 !
310 GOSUB 520 ! INITIALIZE
320 GOSUB 670 ! TEST ROUTINE
330 GOSUB 930 ! DISP ANSWER
340 GOTO 150
350 !
360 ! ************************
370 !
380 !   DISPLAY INSTRUCTIONS
390 !
400 CLEAR
410 DISP " "
420 DISP "INSERT CAP INTO TEST C
    IRCUIT"
430 DISP "TURN ON DMM AND SIGNAL
     GEN."
440 DISP "SET DMM TO 20V AC RANG
    E"
450 DISP " "
460 DISP "     K1 TO RETURN TO M
    ENU"
470 ON KEY# 1 GOTO 150
480 GOTO 470
490 !
500 ! ************************
510 !
520 !
530 ! ************************
540 !
550 !  INITIALIZATION ROUTINE
560 !
570 B=0 ! BEGINING FREQ
580 E=1000000 ! ENDING FREQ
590 V1=1 ! INPUT VOLTAGE(RMS)
600 P=3 1416 ! PIE
610 R=100.608 ! RESISTOR VALUE
620 OUTPUT 717 ;"FU1 FR0HZ AM1VR
    "
630 T=0 ! INITIALIZE COUNTER
640 RETURN
650 !
660 ! ************************
670 !
680 ! ************************
690 !
700 !       TEST ROUTINE
710 !
720 V3= 707*V1
730 F1=B
740 F2=E
750 F3=(F2-F1)/2+F1
760 OUTPUT 717 ;"FR",F3,"HZ"
770 T=T+1 ! INCREMENT COUNTER
775 WAIT 500
780 ENTER 723 ; V2
790 ! V2=V2/1000 ! COMP FOR DMM
800 DISP F3,V2
810 ! IF T>30 THEN 1200 ! CHECK
820 IF V2=V3 THEN 850
830 IF V2>V3 THEN 860
840 IF V2<V3 THEN 880
850 RETURN
860 F1=F3
870 GOTO 750
880 F2=F3
890 GOTO 750
900 !
910 ! ************************
920 !
930 !
940 ! ************************
950 !
960 !      DISPLAY ANSWER
970 !
980 CLEAR
990 C=1/(2*P*P*F3)
1000 BEEP 100,100
1010 BEEP 50,100
1020 BEEP 25,100
1030 C=C*1000000
1040 DISP " "
1050 DISP "CAPACITANCE IN uF IS
     "
1060 DISP USING "3DE" ; C
1070 DISP " "
1080 DISP " "
1090 DISP "   K1 TO RETURN TO ME
     NU"
1100 ON KEY# 1 GOTO 150
1110 GOTO 1100
1120 !
```

LS-4

```
1130 ! ************************
1140 !
1150 !
1160 ! ************************
1170 !
1180 !       ERROR ROUTINE
1190 !
1200 CLEAR
1210 BEEP 1000,100
1220 DISP "TEST NOT FUNCTIONING
     PROPERLY"
1230 DISP " "
1240 DISP "CHECK CIRCUIT AND TRY
      AGAIN"
1250 DISP " "
1260 DISP " "
1270 DISP "K1 TO RETURN TO MENU"
1280 ON KEY# 1 GOTO 150
1290 GOTO 1280
1300 !
1310 ! ************************
1320 END
```

```
950 ! **************************
960 !
970 ! SUBROUTINE TO DETERMINE
980 ! WHICH DMM IS ON THE BUS
985 !
990 ! **************************
995 !
1000 CLEAR
1010 Q1=0
1020 Q2=0
1030 F=0
1040 SET TIMEOUT 7;500
1050 ON TIMEOUT 7 GOTO 1140
1060 S1=SPOLL(704)
1070 Q1=1
1080 DISP "DEVICE 704 IS ON"
1090 F=1
1100 S2=SPOLL(710)
1110 Q2=1
1120 DISP "DEVICE 710 IS ON"
1130 GOTO 1150
1140 ABORTIO 7
1150 IF F=1 THEN GOTO 1170
1160 GOTO 1090
1170 IF Q1=0 AND Q2=0 THEN DISP
     "NO DEVICE ON"
1180 DISP "TEST OVER"
1190 END
```

```
900 ! *************************
910 !
920 ! PROGRAM TO HAVE FLUKE
930 ! 8502A DMM TAKE TEN
940 ! DC VOLTAGE READINGS &
950 ! HAVE THE COMPUTER
960 ! AVERAGE THEM
970 !
980 ! CCTS PROGRAM - B. PERLER
985 !
990 ! *************************
995 !
1000 OUTPUT 704 "V R S5 ?"
1010 FOR I=0 TO 9
1020 OUTPUT 704 "?"
1030 ENTER 704 ; V(I)
1040 DISP "VOLTAGE = ",V(I)
1050 NEXT I
1060 S=0
1070 FOR I=0 TO 9
1080 S=S+V(I)
1090 NEXT I
1100 A=S/10
1105 DISP " "
1110 DISP "AVERAGE = ";A
1115 BEEP
1120 GOTO 5
```

```
900 ! *************************
910 !
920 ! SUBROUTINE TO DETERMINE
930 ! WHICH DMM IS ON THE BUS
940 !
950 ! CCTS LAB - B. PERLER
960 !
970 ! *************************
980 !
990 !
1000 !  ** SERIAL POLL **
1010 Q1=0
1020 Q2=0
1030 F=0
1040 SET TIMEOUT 7;500
1050 ON TIMEOUT 7 GOTO 1120
1060 S1=SPOLL(704)
1070 Q1=1 ! DEVICE 704 IS ON
1080 F=1
1090 S2=SPOLL(710)
1100 Q2=1 ! DEVICE 710 IS ON
1110 GOTO 1130
1120 ABORTIO 7
1130 IF F=1 THEN GOTO 1150
1140 GOTO 1080
1150 IF Q1=0 OR Q2=0 THEN GOTO 1
     170
1160 RETURN
1170 CLEAR
1180 DISP "DMMs NOT TURNED ON"
1190 DISP "TURN ON 8502 AND 8520
     "
1200 DISP " "
1210 DISP "HIT K1 WHEN READY"
1220 ON KEY# 1 GOTO 1240
1230 GOTO 1230
1240 GOTO 1000
1250 RETURN
1260 !  ** DMM TEST ROUTINE **
1270 CLEAR
1280 DISP "SET POWER SUPPLY TO 1
     0VDC"
1290 DISP " "
1300 DISP "HIT K1 WHEN READY"
1310 ON KEY# 1 GOTO 1330
1320 GOTO 1320
1330 OUTPUT 704 ;"?"
1340 ENTER 704 ; V
1350 IF V>=9.99 AND V<=10.01 THE
     N GOTO 1370
1360 GOSUB 1440 ! BAD TEST
1370 DISP " "
1380 DISP "TEST OK, READING IS "
     ;V
1390 DISP " "
1400 DISP "HIT K1 TO CONTINUE"
1410 ON KEY# 1 GOTO 1430
1420 GOTO 1420
1430 RETURN
1440 ! BAD TEST ROUTINE


1450 RETURN
```

LS-6

```
   1 ! **************************
   2 !
   3 ! PROGRAM TO COUNT PARTS
   4 !
   5 ! CCTS LAB EXERCISE SOLUTION
   6 !      BY BRUCE PERLER
   7 !
   8 ! **************************
  10 ! ** MAIN ROUTINE **
  20 GOSUB 500 ! SERIAL POLL
  30 GOSUB 400 ! INITIALIZE
  40 GOSUB 790 ! DMM TEST
  50 CLEAR
  60 DISP "EMPTY SCALE AND HIT K1
     "
  70 ON KEY# 1 GOTO 90
  80 GOTO 80
  90 GOSUB 360 ! TAKE READING
 100 E=V1
 110 CLEAR
 120 DISP "PLACE KNOWN # OF PARTS
     ON SCALE"
 130 DISP "ENTER # OF PARTS AND H
     IT END LINE"
 140 INPUT N
 150 GOSUB 360
 160 K=V1
 170 W=(K-E)/N
 180 CLEAR
 190 DISP "PLACE UNKNOWN # OF PAR
     TS ON SCALE AND HIT K1"
 200 ON KEY# 1 GOTO 220
 210 GOTO 210
 220 GOSUB 360
 230 U=V1
 240 U=U-E
 250 P=U/W
 260 CLEAR
 270 DISP "NUMBER OF PARTS IS"
 280 DISP USING "3D" ; P
 290 DISP " "
 300 DISP "HIT K1 TO REPEAT"
 310 ON KEY# 1 GOTO 50
 320 GOTO 320
 330 END
 340 !
 350 !
 360 !   ** READING ROUTINE **
 370 OUTPUT 710 ;"?"
 380 ENTER 710 ; V1
 390 RETURN
 400 !    ** INITIALIZATION  **
 410 CLEAR 7
 420 CLEAR 704
 430 OUTPUT 704 ;"V R S0"
 440 CLEAR 710
 450 OUTPUT 710 ;"V R0 D1 F3 T1"
 460 V=0
 470 RETURN
 480 !
```

```
490 !
500 !   ** SERIAL POLL **
510 Q1=0
520 Q2=0
530 F=0
540 SET TIMEOUT 7;500
550 ON TIMEOUT 7 GOTO 620
560 S1=SPOLL(704)
570 Q1=1 ! DEVICE 704 IS ON
580 F=1
590 S2=SPOLL(710)
600 Q2=1 ! DEVICE 710 IS ON
610 GOTO 630
620 ABORTIO 7
630 IF F=1 THEN GOTO 650
640 GOTO 580
650 IF Q1=0 OR Q2=0 THEN GOTO 68
    0
560 OFF TIMEOUT 7
670 RETURN
680 CLEAR
690 DISP "DMMs NOT TURNED ON"
700 DISP "TURN ON 8502 AND 8520"
710 DISP " "
720 DISP "HIT K1 WHEN READY"
730 ON KEY# 1 GOTO 750
740 GOTO 740
750 GOTO 500
760 RETURN
770 !
780 !
790 !   ** DMM TEST ROUTINE **
800 CLEAR
810 DISP "SET POWER SUPPLY TO 10
    VDC"
820 DISP " "
830 DISP "HIT K1 WHEN READY"
840 ON KEY# 1 GOTO 860
850 GOTO 850
860 OUTPUT 704 ;"?"
870 ENTER 704 ; V
890 IF V>9.9 AND V<10.1 THEN GOT
    O 900
890 GOTO 970 ! BAD TEST
900 DISP " "
910 DISP "TEST OK, READING IS ";
    V
920 DISP " "
930 DISP "HIT K1 TO CONTINUE"
940 ON KEY# 1 GOTO 960
950 GOTO 950
960 RETURN
970 CLEAR
980 DISP "TEST READING IS ";V;"
    VOLTS"
990 DISP " "
1000 DISP "THIS IS INCORRECT"
1010 DISP "CHECK POWER SUPPLY AN
     D HIT K1"
1020 ON KEY# 1 GOTO 860
1030 GOTO 1030
1040 END
```

LS-7

```
 10 ! THIS PROGRAM DETERMINES
 15 !            THE
 20 ! CAPACITANCE VALUE OF A CAP

 25 GOTO 8600
 29 CLEAR
 30 F=1000
 40 OUTPUT 717 ;"FU1 FR1000HZ"
 50 OUTPUT 717 ;"AM 3VR"
 55 WAIT 1000
 60 ENTER 722 ; V
 70 V=V/1000
 80 DISP "PROGRAM IS RUNNING"
 90 IF V=2.121 THEN 170
100 IF V<2.1 THEN 150
140 GOSUB 4000 ! 1000 HZ STEP.
150 GOSUB 6000 ! 100 HZ STEP.
160 GOSUB 5000 ! 10 HZ STEP.
170 R=100
180 C=1/(2*3 14159*R*F)
190 DISP "CAPACITANCE EQUALS";C
200 END
2072 V=V/1000
2075 DISP "10E4 STEP.V=",V
4000 ! *************************
4010 ! 1000 HZ STEP ; SUB
4020 ! *************************
4030 F=F+1000
4032 IF F>12000 THEN 9500
4035 IF F<0 THEN GOTO 8500
4040 OUTPUT 717 ;"FR",F,"HZ"
4050 ! WAIT 500
4060 ENTER 722 ; V
4062 V=V/1000
4070 IF V<2.1 THEN RETURN ELSE 4
     000
5000 ! *************************
5010 ! 10 HZ STEP ; SUBROUTINE
5020 ! *************************
5030 F=F+10
5035 IF F<0 THEN GOTO 8500
5037 IF F>12000 THEN 9500
5040 OUTPUT 717 ;"FR",F,"HZ"
5050 ! WAIT 500
5060 ENTER 722 ; V
5062 V=V/1000
5064 IF V=2.1 THEN RETURN
5070 IF V<2.1 THEN RETURN ELSE 5
     000
5100 ! *************************
5110 ! 100 HZ STEP ; SUB
5120 ! *************************
6000 F=F-100
6010 OUTPUT 717 ;"FR".F,"HZ"
6015 IF F<0 THEN GOTO 8500
6017 IF F>12000 THEN 9500
6020 ENTER 722 ; V
6030 V=V/1000
6050 IF V>2.1 THEN RETURN ELSE 6
     000
```

```
7970 ! *************************
7980 ! INSTRUCTIONS ;SUBROUTINE
7990 ! *************************
8000 CLEAR
8005 DISP "SET KEITHLY METER ON"
8010 DISP "THE AC, 20 VOLTS "
8020 DISP "SETTING. "
8025 DISP "TURN ON THE HP 3325-A
     "
8040 DISP "CONNECT CIRCUIT    "
8045 DISP "AS SPECIFIED IN THE"
8050 DISP "THE SPEC SHEET.     "
8060 DISP "INSERT CAP WITHIN"
8070 DISP "35.36 MF AND .133MF"
8170 KEY LABEL
8180 GOTO 8180
8500 ! ***********************
8510 ! ***********************
8520 ! FREQUENCY BELOW 0;SUB.
8530 ! ***********************
8535 ! ***********************
8550 CLEAR
8560 DISP "CAPS VALUE IS >35 "
8570 DISP "MICRO F"
8580 END
8585 ! ***********************
8590 ! SET KEYLABELS ; SUB
8595 ! ***********************
8600 CLEAR
8605 DISP "INSTRUCT-PRESS FOR"
8610 DISP "          REQUIRERED"
8620 DISP "          PROCEDURE."
8630 DISP "CIRCUIT -PRESS FOR"
8640 DISP "          CIRCUIT DIAG
     RAM."
8650 DISP "START   -PRESS FOR PR
     O-"
8660 DISP "          GRAM RUN."
9000 ON KEY# 1,"INSTRUCTIONS" GO
     TO 8000
9005 ON KEY# 2,"-CIRCUIT" GOTO 9
     600
9010 ON KEY# 3,"-START" GOTO 29
9020 KEY LABEL
9030 GOTO 9030
9500 ! ***********************
9510 ! ***********************
9520 ! TO HIGH OF FREQUENCY;SUB
9530 ! ***********************
9540 ! ***********************
9550 CLEAR
9560 DISP "CAP VALUE IS LOWER"
9570 DISP "THEN .133 MICRO F "
9580 END
9600 ! *************************
9610 ! CIRCUIT DIAGRAM
9620 ! *************************
9640 DISP "HP+----------------"
9650 DISP "            -   POS "
9660 DISP "          ----- ^   "

9670 DISP "          CAP   VO  "
9680 DISP "          ----- ^   "
9690 DISP "      100   -   NEG "
9720 DISP "HPG--III-----------"
9725 DISP "      RES           "
9730 KEY LABEL
9740 GOTO 9740
```

LS-8

```
10 ! THIS PROGRAM DETERMINES
20 !            THE
30 ! CAPACITANCE VALUE OF A CAP

40 GOTO 800
50 CLEAR
60 F=1000
70 OUTPUT 717 ;"FU1 FR1000HZ"
80 OUTPUT 717 ;"AM 3VR"
90 WAIT 1000
100 OUTPUT 710 ;"VA R2 S5 ?  "
110 ENTER 710 ; V
120 ! V=V/1000
130 DISP "PROGRAM IS RUNNING"
140 IF V=2.1?1 THEN 200
150 IF V<2.1 THEN 170
160 GOSUB 250 ! 10.0 HZ STEP
170 GOSUB 500 ! 100 HZ STEP.
180 GOSUB 360 ! 10 HZ STEP.
190 GOSUB 1160 ! 1 HZ STEP.
200 F=110
210 C=1/(2*3.14159*P*F)
220 DISP "CAPACITANCE EQUALS",C
230 END
240 DISP "10E4 STEP,V=",V
250 ! ************************
260 ! 1000 HZ STEP ; SUB
270 ! ************************
280 F=F+1000
290 IF F<0 THEN GOTO 710
300 OUTPUT 717 ;"FR",F,"HZ"
310 ! WAIT 500
320 OUTPUT 710 ;"?"
330 ENTER 710 ; V
340 ! V=V/1000
350 IF V<2.1 THEN RETURN ELSE 25
    0
360 ! ************************
370 ! 10 HZ STEP ; SUBROUTINE
380 ! ************************
390 F=F+10
400 OUTPUT 717 ;"FR",F,"HZ"
410 ! WAIT 500
420 OUTPUT 710 ;"?"
430 ENTER 710 ; V
440 ! V=V/1000
450 IF V=2.1 THEN RETURN
460 IF V<2.1 THEN RETURN ELSE 36
    0
470 ! **      ******************
480 ! 100     STEP ; SUD
490 ! ***,********************
500 F=F-100
510 OUTPUT 717 ;"FR",F,"HZ"
520 IF F<0 THEN GOTO 710
530 OUTPUT 710 ;"?"
540 ENTER 710 ; V
550 ! V=V/1000
560 IF V<2.1 THEN RETURN ELSE 50
    0
570 IF F=100 THEN GOTO 190
580 ! ************************
590 ! INSTRUCTIONS ;SUBROUTINE
600 ! ************************
610 CLEAR
620 DISP "TURN ON THE HP 3325-A
    "
630 DISP "ALSO,TURN ON THE FLUK
    8520 A."
640 DISP "CONNECT CIRCUIT     "
650 DISP "AS SPECIFIED IN THE"
660 DISP "THE SPEC SHEET.     "
670 DISP "INSERT CAP WITHIN"
680 DISP "20     MF AND.001 MF"
690 KEY LABEL
700 GOTO 700
710 ! ************************
720 ! ************************
730 ! FREQUENCY BELOW 0;SUB.
740 ! ************************
750 ! ************************
760 CLEAR
770 DISP "CAPS VALUE IS >25 "
780 DISP "MICRO F"
790 END
800 CLEAR
810 DISP "INSTRUCT-PRESS FOR"
820 DISP "           REQUIRERED"
830 DISP "           PROCEDURE."
840 DISP "CIRCUIT -PRESS FOR"
850 DISP "           CICUIT DIAGR
    M."
960 DISP "START   -PRESS FOR PR
    -"
970 DISP "           GRAM RUN."
980 ON KEY# 1,"INSTRUCTIONS" GO
    U 610
990 ON KEY# 2,"-CIRCUIT" GOTO 1
    20
900 ON KEY# 3,"-START" GOTO 50
910 KEY LABEL
920 GOTO 920
930 ! ************************
940 ! ************************
950 ! TO HIGH OF FREQUENCY;SUB
960 ! ************************
970 ! ************************
980 CLEAR
990 DISP "CAP VALUE IS LOWER"
1000 DISP "THEN .01  MICRO F "
1010 END
1020 ! ***********************
1030 ! CIRCUIT DIAGRAM
1040 ! ***********************
1050 DISP "HP+----------------"
1060 DISP "           -    POS "
1070 DISP "          -----     "
1080 DISP "           CAP   VO "
1090 DISP "          -----     "
1100 DISP "     100    -   NEG "
```

LS-9

```
1110 DISP "HPG--III-----------"
1120 DISP "      RES           "
1130 KEY LABEL
1140 GOTO 1140
1150 ! ************************

1160 ! 1 HZ STEP ; SUBROUTINE.
1170 ! ************************
1180 F=F-1
1190 IF F<150 THEN WAIT 1000
1200 OUTPUT 717 ;"FR",F,"HZ"
1210 OUTPUT 710 ;"?"
1220 ENTER 710 ; V
1230 ! V=V/1000
1240 IF V=2.1 THEN RETURN
1250 IF V>2.1 THEN RETURN ELSE 1
     160
```

LS-10

```
10 DISP "THIS PROGRAM DETERMINES THE VALUE OF A CAPACITOR ";
20 DISP "BETWEEN 2 MICROFARADS AND 1000 PICOFARADS." ! DEFINES PROGRAM
30 ON KEY# 1,"CIRCUIT" GOSUB 430 ! CIRCUIT DIAGRAM
40 ON KEY# 2,"YES" GOTO 10 ! RERUN PROGRAM
50 ON KEY# 3,"NO" GOTO 320 ! END PROGRAM
60 ON KEY# 7,"RUN TEST" GOSUB 330 ! RUN
70 KEY LABEL ! DISPLAYS LABEL
80 WAIT 3000 ! WAIT 3 SEC
90 CAPX: ! DETERMINE CAP VALUE
100 R=100 @ N=24500 @ AP=1 ! DEFINE VARIABLES
110 OUTPUT 723 ;"T3" ! TRIGGER DMM
120 ENTER 723 ; VRD ! ENTER VOLTAGE READING FROM DMM
130 IF VRD<= .7075 AND VRD>= .7065 THEN 160 ELSE 140 ! DEFINES BOUNDARIES
140 IF VRD<.7065 THEN 240 ! PREPARE TO CHANGE FREQ
150 IF VRD>.7075 THEN 210 ! PREPARE TO CHANGE FREQ
160 CT=1/(2*PI *R*N) ! DETERMINE INTERMEDIATE  CAP VALUE
170 CX=CT/10000 ! DETERMINE CAP VALUE FOR DISPLAY
180 IF N<100 OR N>1600000 THEN 520 ! TOLERANCES FOR CAPS WITHIN THESE FREQS
190 DISP "CAP VALUE IS ";CX ! PRINT CAP VALUE
200 GOTO 270 ! RUN AGAIN ?
210 N=N*1.9 ! INCREASE FREQ
220 OUTPUT 707 ;"FR";N;"HZ" ! SEND FREQ CHANGE TO FG
230 GOTO 110 ! RETURN TO EVALUATION
240 N=N/2 ! DECREASE FREQ
250 OUTPUT 707 ;"FR";N;"HZ" ! SEND FREQ CHANGE TO FG
260 GOTO 110 ! RETURN TO DO NEW EVALUATION
270 AGAIN: ! SUB ROUTINE FOR RUN AGAIN ?
280 DISP "TO DO ANOTHER OR STOP ";
290 DISP "PRESS K2 FOR YES OR K3 FOR NO" ! GO AGAIN OR NOT
300 WAIT 5000 ! WAIT 5 SEC
310 WAIT 5000 ! WAIT 5 SEC
320 DISP "THANK YOU FOR YOUR PATIENCE" @ END ! PROGRAM OVER
330 SETUP: ! SUB RTN FOR SETTING FG & DMM
340 CLEAR ! CLEAR SCREEN
350 RESET 7 ! RESETS INTERFACE #7
360 CLEAR 7 ! DEVICES TO DEFAULT
370 REMOTE 723 ! TO BUS CONTROL
380 LOCAL LOCKOUT 7 ! FRONT DISABLE
390 OUTPUT 723 ;"H2" ! AC VOLTS TRIGGER ONCE
400 REMOTE 707 ! TO BUS CONTROL
410 OUTPUT 707 ;"PR FU1 AP1.0VO FR24.5KZ" ! 1V P-P SINE AT 24.5KHZ
420 RETURN ! RETURN TO MAIN PROGRAM
430 CIRCUIT: ! CIRCUIT DIAGRAM SUB ROUTINE
440 CLEAR ! CLEAR SCREEN
450 DISP "FG SIGNAL OUT------RES--------|--------DMM SIGNAL IN"
460 DISP "                  100 OHMS  __|__"
470 DISP "                              Cx" ! DRAW DIAGRAM
480 DISP "                            --|--"
490 DISP "FG GRND-----------------------|--------DMM GRND"
500 WAIT 8000 ! WAIT 8 SEC
510 RETURN ! RETURN TO MAIN PROGRAM
520 ILLEGAL: ! SUBROUTINE FOR MISTAKE
530 DISP "INCORRECT CAPACITOR SIZE OR IMPROPER CIRCUIT" ! CHECK COMPONENTS
540 GOTO 290 ! TRY AGAIN
```

LS-11

```
10 CLEAR ! CLEARS SCREEN
20 DISP "THIS PROGRAM UTILIZES A LOAD CELL TO WEIGH PARTS AND DETERMINE THE COU
T"
30 DISP " OR WEIGH ITEMS BETWEEN 1 GRAM AND 1 KILOGRAM." ! DEFINES PROGRAM
40 CLEAR 7 ! DEVICES TO DEFAULT
50 REMOTE 7 @ S=7 ! TO BUS CONTROL AND SET BUS 7 = TO VARIABLE S
60 M=18 @ N=23 ! SET VARIABLES FOR ADDRESSES OF DMM'S
70 S1=SPOLL (S*100+M) ! SERIAL POLL 718
80 SET TIMEOUT S;500 ! TIMEOUT = 500 MILLISEC
90 ON TIMEOUT S GOTO 120 ! DEVICE 718 NOT ON BUS
100 DISP "FLUKE 8502A ON BUS. USE THIS DMM IN CIRCUIT" ! MSG FOR OPERATOR
110 P$="S1" @ GOTO 160 ! SET STRING VARIABLE FOR 718
120 DISP "8502A NOT PRESENT" ! MSG RE:718
130 OFF TIMEOUT S ! REMOVE TIMEOUT CMD
140 WAIT 3000 ! 3 SEC WAIT
150 S2=SPOLL (S*100+N) ! SERIAL POLL 723
160 SET TIMEOUT S;500 ! SET TIMEOUT FOR 723
170 ON TIMEOUT S GOTO 430 ! ABORT 723
180 OFF TIMEOUT S ! TURN OFF TIMEOUT CLOCK
190 DISP "HP 3478A ON BUS" ! MSG TO OPERATOR
200 WAIT 3000 ! 3 SEC WAIT
210 Q$="S2" ! SET VARIABLE TO 723
220 IF Q$="S2" AND P$="S1" THEN 230 ELSE 290 ! CHECK IF BOTH ON BUS
230 DISP "BOTH DMM's ON BUS. USE 8502A IN CIRCUIT" ! MSG TO OPERATOR
240 PRINTER IS 705 ! ADDRESS OF PRINTER
250 PRINT "8502A ADDRESS IS 718" ! DISPLAYS DMM ADDRESS
260 WAIT 3000 ! 3 SEC WAIT
270 !
280 !
290 ! GOSUB SETUP OR WEIGHING OR COUNTING
300 ON KEY# 1,"PWRUP TEST" GOSUB 500
310 ON KEY# 3,"METRIC WT" GOSUB 670
320 ON KEY# 5,"PARTS CNTR" GOSUB 1040
330 KEY LABEL ! DISPLAYS LABELS
340 DISP "PRESS K1 FOR POWER UP SELF-TEST" @ DISP @ DISP ! OPERATOR PROMPT
350 BEEP ! AUDIO FOR WARNING MESSAGE
360 DISP "WARNING !!!  POWER UP MUST BE MADE ON INITIAL RUN" ! WARNING MSG
370 DISP @ DISP ! SPACES BTWN MSGS
380 DISP "PRESS K3 FOR METRIC WEIGHING" @ DISP @ DISP ! OPERATOR PROMPT
390 DISP "PRESS K5 FOR PARTS COUNTING" ! OPERATOR PROMPT
400 GOTO 400 ! HOLD
410 !
420 !
430 NOGO: ! SUBROUTINE FOR DMM'S OFF BUS
440 ABORTIO 7 ! STOP ALL TRAFFIC ON BUS
450 OFF TIMEOUT S ! STOP ALL TIMEOUTS ON BUS
460 DISP "NEITHER 8502A OR 3478A ON LINE. CHECK POWER TO DMM's." ! ERROR MSG
470 END ! END PROGRAM
480 !
490 !
500 SETUP: ! SUBROUTINE FOR DMM AND GPIO INITIALIZATION
510 RESET 7 ! RESETS INTERFACE #7
520 CLEAR 7 ! DEVICES TO DEFAULT
530 REMOTE 718,723 ! TO BUS CONTROL
540 LOCAL LOCKOUT 7 ! FRONT DISABLE
550 OUTPUT 723 "H1" ! DC, AUTO RANGE, ONE TRIGGER
560 OUTPUT 718 "VR0S7," ! DC, 312.5MV, 128 SAMPLES/READING
570 CLEAR 4 ! DEVICES TO DEFAULT
580 CONTROL 4,5 ; 1 ! REGISTER 5, PRIMARY ADDRESS PORT B
590 CONTROL 4,5 ; 5 ! REGISTER 5, PRIMARY ADDRESS PORT D
```

LS-12

```
600 CONTROL 4,4 ; 194 ! REGISTER 4, NO HANDSHAKE, INVERT PORT B DATA
610 CONTROL 4,8 ; 2 ! REGISTER 8, ENABLE PORT B OUTPUTS
620 GOSUB 1410
630 GOSUB 1410
640 GOTO 290
650  !
660  !
670 ! SUBROUTINE FOR METRIC WEIGHING
680 CLEAR ! CLEAR SCREEN
690 DISP " SCALE MUST BE EMPTY FOR INITIAL READING" ! MSG TO OPERATOR
700 GOSUB 1520
710 RESET 7 ! RESETS INTERFACE #7
720 CLEAR 7 ! DEVICES TO DEFAULT
730 A$="VROS7,$?" ! SET STRING VARIABLE FOR DMM OUTPUT
740 OUTPUT 718 "VROS7,$?" ! DC, 312.5MV, 128 SAMPLES/READING, CLEAR, TRIGGER
750 OUTPUT 718 USING "#,8A" ; A$ ! TAKE READING
760 WAIT 500 ! 500 MSEC WAIT FOR SETTLING TIME (IMPERATIVE)
770 ENTER 718 ; VE ! READ DATA
780 DISP "EMPTY READING IS ";VE @ DISP @ DISP ! SHOW READING THEN SPACE LINE
790 DISP "PLACE 1 GM WT ON SCALE TO CALIBRATE" ! PROMPT OPERATOR
800 GOSUB 1520
810 RESET 7 ! RESETS INTERFACE #7
820 CLEAR 7 ! DEVICES TO DEFAULT
830 OUTPUT 718 "VROS7,$?" ! SETUP  DMM
840 OUTPUT 718 USING "#,8A" ; A$ ! TRIGGER DMM
850 WAIT 500 ! SETTLING TIME
860 ENTER 718 ; VA ! RECORD READING
870 B1=.000005 ! REQUIRED DUE TO MIN RESOLUTION OF DMM 10 MICROVOLTS
875 DISP "1 GM READING IS ";VA @ DISP @ DISP ! SHOW READING THEN LINE SPACES
880 DISP "1 GM READING IS ";VA @ DISP @ DISP
890 DISP "PLACE ANY METRIC WT BTWN 1 GM & 1 KG ON SCALE" ! PROMPT OPERATOR
900 GOSUB 1520
910 RESET 7 ! RESETS INTERFACE #7
920 CLEAR 7 ! DEVICES TO DEFAULT
930 OUTPUT 718 USING "#,8A" ; A$ ! TRIGGER DMM
940 WAIT 500 ! SETTLING TIME
950 ENTER 718 ; VB@ DISP "VB= ";VB ! TAKE READING AND DISPLAY
960 B2=VB-VE ! SUBTRACT 1ST READING FM FINAL READING
970 WT=ABS (B2/B1) ! COMPUTE TOTAL GRAMS
980 DISP "TOTAL WT IN GRAMS IS ";WT
990 PRINTER IS 705 ! DEFINE PRINTER ADDRESS
1000 PRINT "TOTAL WT IN GRAMS IS ";WT ! PRINTOUT TOTAL GRAMS
1010 END ! END OF METRIC PROGRAM
1020 !
1030 !
1040 ! SUBROUTINE FOR PARTS COUNTER
1050 CLEAR ! CLEAR SCREEN
1060 DISP "LET DMM TAKE READING OF EMPTY SCALE" ! MSG TO OPERATOR
1070 GOSUB 1520
1080 RESET 7 @ CLEAR 7 ! RESETS INTERFACE #7 ! DEVICES TO DEFAULT
1090 A$="VROS7,$?" ! DEFINE DMM SETUP
1100 OUTPUT 718 "VROS7,$?" ! TRIGGER DMM
1110 OUTPUT 718 USING "#,8A" ; A$ ! TAKE READING
1120 WAIT 500 ! SETTLING TIME
1130 ENTER 718 ; VC ! RECORD READING
1140 DISP "EMPTY READING IS ";VC ! DISPLAY READING
1150 DISP "WEIGH ONE ITEM" ! MSG TO OPERATOR
1160 GOSUB 1520
1170 RESET 7 @ CLEAR 7 ! RESET INTERFACE #7 ! DEVICES TO DEFAULT
1180 OUTPUT 718 "VROS7,$?" ! DEFINE OUTPUT OF DMM
```

LS-13

```
1190 OUTPUT 718 USING "#,8A" ; A$ ! TRIGGER DMM
1200 WAIT 500 ! SETTLING TIME
1210 ENTER 718 ; VA ! RECORD READING
1220 DISP "1 ITEM READING IS ";VA ! DISPLAY READING
1230 DISP "PLACE REMAINING ITEMS ON SCALE" ! MSG TO OPERATOR
1240 GOSUB 1520
1250 RESET 7 @ CLEAR 7 ! RESET INTERFACE #7 ! DEVICES TO DEFAULT
1260 OUTPUT 718 "VROS7,$?" ! SETUP DMM
1270 OUTPUT 718 USING "#,8A" ; A$ ! TRIGGER DMM
1280 WAIT 500 ! SETTLING TIME
1290 ENTER 718 ; VB ! RECORD READING
1300 DISP "RMNG ITEMS READING IS ";VB ! DISPLAY READING
1310 VX=VA-VC ! DETERMINE 1 ITEM WEIGHT
1320 VY=VB-VA ! DETERMINE REMAINING PARTS WEIGHT
1330 VT=ABS (VY/VX) ! DETERMINE NO. OF PARTS
1340 DISP "NO. OF ITEMS IS ";VT ! DISPLAY NO. OF PARTS
1350 PRINTER IS 705 ! DEFINE PRINTER ADDRESS
1360 PRINT "NO. OF ITEMS IS ";VT ! PRINTOUT
1370 END ! END OF PARTS COUNTING PROGRAM
1380 !
1390 !
1400 ! SUBROUTINE TO STROBE POWER SUPPLY
1410 OUTPUT 401 USING "#,B" ; 10 ! DATA: TENS DIGIT=1, ONES DIGIT=0
1420 OUTPUT 405 USING "#,B" ; 1 ! LATCHES "A" REGISTER
1430 OUTPUT 401 USING "#,B" ; 0 ! DATA 0.1 AND 0.01 IS 0
1440 OUTPUT 405 USING "#,B" ; 2 ! LATCHES "B" REGISTER
1450 OUTPUT 401 USING "#,B" ; 17 ! DATA 40 MA LIMIT, 65 VOLT RANGE
1460 OUTPUT 405 USING "#,B" ; 4 ! LATCHES "C" REGISTER
1470 OUTPUT 405 USING "#,B" ; 8 ! STROBES ALL DATA INTO 4250A
1480 RETURN ! RETURN TO MAIN PROGRAM
1490 !
1500 !
1510 ! SUBROUTINE TO WAIT AS OPERATOR ADDS OR REMOVES ITEMS FROM SCALE
1520 DISP
1530 DISP "HIT END LINE TO CONT " ! MSG TO OPERATOR
1540 INPUT G$ ! WAIT STATE
1550 RETURN ! RETURN TO MAIN PROGRAM
```

LS-14

```
 10 ! *************************
 20 ! * SCALE PROGRAM **********
 30 ! *************************
 40 CLEAR
 50 DISP "PROGRAM IS RUNNING"
 60 GOSUB 240 ! SET TIMEOUT
 70 OFF TIMEOUT 7
 80 T=100*7+I
 90 DISP "DEVICE";T;"BEING USED"
100 GOSUB 1430 ! VOLTAGE SETUP
110 DISP "PRESS K1 FOR GRAMS"
120 DISP "PRESS K2 FOR PARTS"
130 ON KEY# 2 GOTO 180
140 ON KEY# 1 GOTO 160
150 GOTO 130
160 GOSUB 980
170 GOTO 200
180 !
190 GOSUB 400
200 END
210 ! ************************
220 ! ****** SPOLL ROUTINE*****
230 ! ************************
240 FOR I=4 TO 10 STEP 6
250 S=7 @ !
260 SET TIMEOUT S;500
270 ON TIMEOUT S GOTO 320
280 S1=SPOLL(S*100+I)
290 RETURN
300 NEXT I
310 STOP
320 ABORTIO 7
330 PRINT "DEVICE";I;" NOT PRESE
    NT"
340 GOTO 300
350 OFF TIMEOUT 7
360 RETURN
370 ! ************************
380 ! ** PARTS SUBROUTINE *****
390 ! ************************
400 DISP "PARTS WEIGHING PROGRAM
    "
410 DISP "EMPTY SCALE"
420 DISP "PRESS K1 WHEN READY"
430 ON KEY# 1 GOTO 450
440 GOTO 430
450 GOSUB 840
460 WAIT 500
470 ENTER T ; C
480 GOSUB 840
490 WAIT 500
500 ENTER T ; C
510 DISP "PUT KNOWN # OF PARTS O
    N THE SCALE"
520 DISP "ENTER # OF PARTS"
530 INPUT P
540 GOSUB 840
550 WAIT 500
560 ENTER T ; O
570 GOSUB 840
580 WAIT 500
590 ENTER T ; O
600 DISP "PUT UNKNOWN # OF PART
    ON SCALE"
610 DISP "PRESS K1 WHEN READY"
620 ON KEY# 1 GOTO 640
630 GOTO 620
640 GOSUB 840
650 WAIT 500
660 ENTER T ; U
670 GOSUB 840
680 WAIT 500
690 ENTER T ; U
700 G=(O-C)/P
710 F=(U-C)/G
720 DISP "THE # OF PART IS"
730 DISP USING "4D" ; F
740 DISP "PRESS K1 FOR MORE OR K
    2 FOR END"
750 ON KEY# 1 GOTO 780
760 ON KEY# 2 GOTO 790
770 GOTO 750
780 GOTO 600
790 RETURN
800 ! ************************
810   ************************
820 ! * SET UP METERS ********
830 ! ************************
840 ! ************************
850 CLEAR
860 IF T=710 THEN GOTO 900
870 OUTPUT 704 ;"VRS5?"
880 OFF TIMEOUT 7
890 RETURN
900 OFF TIMEOUT 7
910 OUTPUT 710 ;"VR7S5  ?"
920 RETURN
930 ! ************************
940 ! ************************
950 ! *** GRAMS SUBROUTINE*****
960 ! ************************
970 ! ************************
980 DISP "GRAMS WEIGHT PROGRAM"
990 DISP "EMPTY SCALE"
1000 DISP "PRESS K1 WHEN READY"
1010 ON KEY# 1 GOTO 1030
1020 GOTO 1010
1030 GOSUB 800
1040 WAIT 500
1050 ENTER T ; E
1060 GOSUB 800
1070 WAIT 500
1080 ENTER T ; E
1090 DISP "PUT ON KNOWN # OF GRA
     MS"
1100 DISP "ENTER # OF GRAMS"
1110 INPUT O
1120 GOSUB 800
1130 WAIT 500
1140 ENTER T ; G
```

```
1150 GOSUB 800
1160 WAIT 500
1170 ENTER T ; G
1180 DISP "PUT ON UNKNOWN # OF G
     RAMS"
1190 DISP "HIT K1 WHEN READY"
1200 ON KEY# 1 GOTO 1220
1210 GOTO 1200
1220 GOSUB 800
1230 WAIT 500
1240 ENTER T ; G1
1250 GOSUB 800
1260 WAIT 500
1270 ENTER T ; G1
1280 L=(G-E)/Q
1290 K=(G1-E)/L
1300 DISP "THE # OF GRAMS IS"
1310 DISP USING "4D" ; K
1320 DISP "HIT K1 FOR MORE OR K2
      FOR END"
1330 ON KEY# 1 GOTO 1180
1340 ON KEY# 2 GOTO 1360
1350 GOTO 1330
1360 RETURN
1370 ! ************************
1380 ! ************************
1390 ! ***INPUT VOLT CHECK*****
1400 ! ************************
1410 ! ************************
1420 CLEAR
1430 DISP "APPLY SCALE REFERECE
     VOLTAGE TO THE METER BEING
     USED"
1440 DISP "PRESS K1 WHEN READY"
1450 ON KEY# 1 GOTO 1470
1460 GOTO 1450
1470 GOSUB 800
1480 WAIT 500
1490 ENTER T ; V
1500 GOSUB 800
1510 WAIT 500
1520 ENTER T ; V
1530 IF V>10.05 OR V<9.9 THEN GO
     TO 1560
1540 DISP "VOLTAGE IS WITHIN LIM
     ITS,HOOK BACK UP TO THE SCA
     LE"
1550 RETURN
1560 DISP "VOLTAGE IS NOT WITHIN
      LIMITS  REAJUST AND TRY AG
     AIN."
1570 WAIT 2000
1580 GOTO 1420
```

LS-16

APPENDIX B
REFERENCE & BIBLIOGRAPHY

Abenaim, D. (Electronic Instrumentation Division, GenRad Inc., Concord, MA). "Impactof Microprocessors on Design of Analog Instrumentation." **Electro 76 Professional Program, Boston, May 11-14,1976.**

Ableidinger, A. (Tektronix, Inc., Beaverton, OR). "Unraveling the Mystery of the GPIB."**Tekscope** 10 (1978): 3-6

Allen, D. "A Minifloppy Interface".**BYTE**, February 1978, pp. 114-25

Anundson, Robert L. (Tektronix, Inc., Beaverton, OR). "2nd-Generation GPIB Equipment Offers Easy-to-Use Improvements".**EDN** August 19,1981, pp165-72.

Baunach, S.C. (Tektronix, Inc., Beaverton, OR). "Design Advantages and Limitations in Connecting Computational and Readout Equipment to the GPIB". 1976 **WESCON** Professional Program, Los Angeles, September 14-17, 1976.

Bond, J. "Computer-Controlled Instruments Challenge Designer's Ingenuity", **EDN**, March 5, 1974, pp 32-37.

Buchanan, G. (Texas Instruments Ltd., Bedford, England). "Implementation of the Bus on a Chip". 1980 Electro Professional Program, Electronic Conventions, El Segundo, CA 90245.

Burhans, R.W. "A Simpler Digital Cassette Tape Interface". **BYTE**, October 1978, pp 142-43.

Ciarcia, S. "No Power for Your Interfaces? Build a 5W DC to DC Converter". **BYTE**, October 1978, pp 26-31.

-. "A Penny Pinching Address State Analyzer". **BYTE**, February 1978, pp 6-12.

Coats, T. "Interfacing to the Interface: Practical Considerations beyond the Scope of IEEE Standard 488". WESCON 1975, Session 3.

Comerford, R.W. (Test, Measurement and Control Editor). "Measurement Science is Catching Up".**Electronics** July 31, 1980, pp 86-87.

Corbridge, C. (Tektronix, Inc., Beaverton, OR). "The Use of the IEEE 488 Bus in Meteorology and Calibration". 1980 Electro Professional Program, Electronic Conventions, El Segundo, CA 90245.

Corson, D. (Hewlett-Packard Co.). "The Logic Analyzer In GPIB Systems". **Electronic Engineering Times**, July 23, 1980, p 43.

Cram, Robert S. "Standard Codes and Format for the IEEE 488 Bus". In **Proceedings, ATE Seminar/Exhibit**, p V-31. Boston: Benwill Publishing Corp., January 1981.

Dietrich, H.E. "Visualizing Interface Bus Activity". **Hewlett-Packard Journal**, January 1975, pp 19-23.

Donahoo, P.T. (John Fluke Mfg. Co., Mountlake Terrace. WA). "What to Look for in a Manufacturer's IEEE-488 Specifications". Electro Professional Program, Electronic Conventions, El Segundo, CA 90245.

"Instruments '80 Benchtoppers-Both Digital and Analog-Change Their Looks to Solve New Problems". **Electronic Design** 21 (October 11, 1979): 102-106.

"Electro/80". **Electronics**, April 24, 1980.

Evans, J. and Merrill, H. "A Standsard Interface Applied to Measurement Systems". In **ISA Proceedings, 22nd International Instruments Symposium**, Vol. 3, 1976, pp 17-28.

Everett, C.M. (Dana Laboratories Inc., Irvine, CA). "Design and Application Restraints in the Implementation of Smart Instrumentation". Electro 76 Professional Program, May 11-14, 1976.

Feinberg, P. (Wavetek Rockland, Inc.). "Spectrum Analyzers Adapt to Diverse Tasks". **Electronic Design**, October 11, 1980, pp 199-205.

Fisher, Eugene R. (Affiliate Faculty in Electrical Engineering, Colorado State University). **The IEEE 488 Bus**, a 12-cassette videotaped course. (Engineering Renewal and Growth Program, CSU, Fort Collins, CO.,1980).

-. (Lawrence Livermore National Laboratory. Livermore, CA). "The System Designer Looks at the GPIB". 1976 WESCON Professional Program, Los Angeles, September 14-17, 1976.

Fischer, Eugene R., and Jensen, C. William. **Study Guide: An Introduction to the IEEE 488 Bus-From Chips to Debug.** Engineering Renewal and Growth Program, Colorado State University, Fort Collins, CO, 1980.

-. **PET/CBM and the IEEE 488 Bus (GPIB)**, 2nd Ed. Osborne/McGraw-Hill, Berkeley, 1982.

Fisher, P.D., and Welch, S.M. "Part 2: How to Build an Automated System Around a Programmable Calculator". **Electronics**, May 16, 1974, pp 104-109.

Fluke J.M., Jr. (John Fluke Manufacturing Co., Inc., Mountlake Terrace, WA). "System Considerations in Using the IEEE Digital Instrument Bus". WESCON 1975, Session 3.

-. **IEEE Standard 488 1975 Digital Interface for Programmable Instrumentation.** Applications Bulletin, Fluke AB-36.

-. "What Makes an Instrument Truly System Compatable?". **EE/Systems Engineering Today**, May 1973.

Fluke, John Manufacturing Co., Inc. "Technical Bulletin, Troubleshooting Information". **John Fluke Mfg. Publication C0076A.**

Freytag, H.H., and teh-Thiede, A.K. "Konzeption Programmierbarer Messgerate mit IEC-Standard-Interface". **Frequenz**, July 30, 1976, pp 190-95.

Frisch, Arnie, and Hollister, Allen "GPIB replacement coordinates hardware, simplifies software". **Electronic Design**, December 12, 1985, pp 107-12.

Geisman, J. "IEEE-488: User Fundamentals". 1980 Electro Professional Program, Electronic Conventions, El Segundo, CA 90245.

Gifford, Tom "Many Choices Complicate IEEE-488 Instrument Selection". **Electronic Products**, March 1, 1985, pp

Graube, M. (Tektronix Inc., Beaverton, OR). "IEEE 488 Bus (GPIB): Present Status and Future Direction". 1980 Electro Professional Program, Electronic Conventions, El Segundo, CA 90245.

—. "IEEE-488 Case Histories: Progress and Problems". 1980 Electro Professional Program, Electronic Conventions, El Segundo, CA 90245.

Halissy, B. (Hewlett-Packard Co., Fort Collins, CO). "Directions in GPIB Controllers". 1980 Electro Professional Program, Electronic Conventions, El Segundo, CA 90245.

Harold, Peter "New IEEE-488 Bus Specs Promice to Ease Systems Integration and Enhance Performance". **EDN**, September 19, 1985.

Hewlett-Packard Co. "Calculator Control of the 8660A/C Synthesized Signal Generator". **Hewlett-Packard Publication** AN 164-2.

—. "Recent Advances in Pulsed RF and Microwave Frequency Measurements". **Hewlett-Packard Publication** AN 173.

—. "Measuring Linearity of VCOs from 10Hz to 23GHz". **Hewlett-Packard Publication** AN 181-1

—. "Data Acquisition with the 5200B Measuring System". **Hewlett-Packard Publication** AN 181-2.

—. "A Common Digital Interface for Programmable Instruments: The Evolution of a System". **Hewlett-Packard Journal.** October 1972, pp 8-11.

—. **Condensed Description of the Hewlett-Packard Interface Bus.** Loveland, CO: Hewlett-Packard Co., March 1975.

—. "Configuration of a 2-18.6GHz Synthesized Frequency Source Using the 8620C Sweep Oscillator". **Hewlett-Packard Publication** AN 187-2.

—. "Three HP-IB Configursations for Making Microwave Scalar Measurements". **Hewlett-Packard Publication** AN 187-3.

—. "Calculator Control of the 8620C Sweep Oscillator Using the HP-IB". **Hewlett-Packard Publication** AN 187-5.

—. "Automatic Measurements Using the HP 436A MicrowAve Power Meter". **Hewlett-Packard Publication** AN 196.

—. "Routine QA Measurements of Precision Resistors". **Hewlett-Packard Publication** AN 201-1.

-. "Measuring DifferentiAl Non-linearity of a Voltage-Controlled Oscillator". **Hewlett-Packard Publication** AN 202-2.

-. "A Multiple Station Electronic Test System". **Hewlett-Packard Publication** AN 201-3.

-. "Performance Evaluation of HP-IB Using RTE Operating Systems". **Hewlett-Packard Publication** AN 201-4.

-. "Measuring Wide-Band Noise with the HP 3045A Automatic Spectrum Analyzer". Hewlett-Packard Publication AN 201-6.

-. "HP 1000/HP-IB: High Performance Software for the HP 3455/3495A Subsystem". **Hewlett-Packard** AN 201-7.

-. "Applications and Performance of the 8671A/8672A Microwave Synthesizers". **Hewlett-Packard** AN 218-1.

-. "Obtaining Millihertz Resolution from the 8671A and 8672A". **Hewlett-Packard** AN 218-2.

-. "A 1 MHz-18GHz Signal Generator with 1, 2, or 3 Hz Resolution". **Hewlett-Packard Publication** AN 218-3.

-. "Semi-Automatic Measurements Using the 8410B Microwave Network Analyzer and the 9825A Desktop Computer". **Hewlett-Packard Publication** AN 221.

-. "Signal Conditioning: HP 22914A Breadboard Card". **Hewlett-Packard Publication** AN 224-2.

-. "HP-IB Power Supply Interface Guide". **Hewlett-Packard Publication** AN 250-1.

-. "An Example of Automatic Measurement of Conducted EMI with the HP 8568A Spectrum Analyzer". **Hewlett-Packard Publication** AN 270-1.

-. "Adding Soft Copy Graphics to 9825A Based HP-IB Systems Using the Model 1350A Graphic Translator". **Hewlett-Packard Publication** AN 271-1.

-. Monitoring the IEEE-488 Bus with the 1602A Logic State Analyzer". **Hewlett-Packard Publication** AN 280-1.

-. "The 1602A Logic State Analyzeras an Automatic Test Instrument". **Hewlett-Packard Publication** AN 280-4.

-. "Making Measurements on Wide Buses with HP's Model 1602A LOgic State Analyzers". Hewlett-Packard Publication AN 280-4.

-. "8662A RF Synthesized Signal Generator Performance, Familiarization, and applications". **Hewlett-Packard Publication** AN 283-1.

-. "Applications and Operation of the 8901A Modulation Analyzer". **Hewlett-Packard Publication** AN 287-1.

-. "Waveform Analysis Using the 5328A Modulation Analyzer". **Hewlett-Packard Publication** AN 286-1.

–. "Hewlett-Packard Interface Bus User's Guide". **Hewlett-Packard Publication 9830A.**

–. "HP-IB, Improving Measurements in Engineering and Manufacturing-A Collection of Useful Technical Information". **Hewlett-Packard Publication 5952-0058.**

–. "Tutorial Description of the Hewlett-Packard Interface Bus". **Hewlett-Packard Publication 5952-0156** NOTE: Most comprehensive single publication on the GPIB.

–. "Application Guide for the 3325A Synthesizer/Function Generator - Product Note 3325A-1". **Hewlett-Packard Publication 5953-5120.**

–. "Operators introduction to the 3314A - Product Note 3314A-1". Hewlett-Packard Publication 5953-5102.

–. "Using the 3478A - Product Note 3478A-1". **Hewlett-Packard Publication 5953-6901.**

Hordeski, M. "Balancing P-Interface Tradeoffs". **Digital Design**, April 1977, pp 66-72.

Huang, I.Y. "An Interconnect Standard". **Electronic Engineering Times**, September 27, 1976, p 14.

Institute of Electrical and Electronics Engineers, Inc. **The IEEE Standard Digital Interface for Programmable Instrumentation.** IEEE Std 488-1975, IEEE Std 488-1975-ANSI MC 1.1-1975. New York: IEEE, 1975.

–. **IEEE Standard Digital Interface for Programmable Instrumentation.** IEEE Std 488-1978 (Revision of ANSI/IEEE Std 488-1975). New York: IEEE, 1978. NOTE: Document defining GPIB.

Intel Corporation. **Peripheral Design Handbook.** Santa Clara, CA, February 1979.

Kimball, J. (Tektronix, Inc., Beaverton, OR). "The IEEE 488 Bus-Going Your Way?" **Tekscope** 10 (1987):7-10.

Klaus, J. "Wie Funktioniert der IEC-Bus?" **Electronik** 10 (1977):63-74.

Knoblock, D.E.; Loughry, D.C.; and Vissers, C.A. "Insight into Interfacing". **IEEE Spectrum** 12 (1975): 50-57.

Koch, P.N. (Texas Instruments). "Talking Meter Makes Eyes-Free Measurements". **Electronic Design,** October 11, 1980, pp 185-89.

Kubota, S. and Kruger, C. (Interface Technology). "IEEE 488 Bus Testing: One Manufacturer's Viewpoint". 1980 Electro Professional Program, Electronic Conventions, El Segundo, CA 90245.

Kwaspen, Mathieu (Eindhoven University of Technology. Netherlands). "488-Bus Interface Eases Troubleshooting". **EDN**, August 5, 1981, pp 150-52.

Laengrich, Norbert (Racal-Dana Instruments, Inc., Irvine, CA 927133). "Instrument Intelligence Determines 488 Bus Speed". **Electronic Design**, October 15, 1981, pp 181-86.

-. (Dana Laboratories Inc., Irvine, CA). "The IEEE Std 488/1975-General-Purpose Means of Providing Measurement and Stimulus Instrument communication". 1976 WESCON Professional Program, Los Angeles, September 14-17.

Laroff, G.P. (Tektronix, Inc., Boston, MA). "Tektronix GPIB Applications Support". **Publication 006-2519-00**, 1976.

Lee, R.C. "Microprocessor Implementation of a Measurement Instrument and its Interface". WESCON 1975, San Francisco, Session 3.

Leibson, S. (Hewlett-Packard Co.). "The Standard Interface". **Keyboard**, July 1979, pp 8-11.

-. "What are RS-232C and IEEE 488?". **Instruments and Comtrol Systems**, January 1980, pp 47-53.

Levine, S.T. "Instrument Interfacing Can Be Done-But It Takes a Bit of Doing". **Electronic Design** 24 (1973), pp 74-79.

Lewin, L.M., and Helm, G. "A Versatile Microcomputer I/O Interface". **Behavior Research Methods and Instrumentation** 11 (1979): 445-46.

Lipovac, V. "Design an IEEE-488 Bus into an FPLA and Speed System Operation. An Extra Benefit: The Technique Allows You to Interface Any Number of Programmable Instruments". **Electronic Design** 24 (1977): 104-11.

Loughry, D.C. "The Hewlett-Packard Interface Bus: Current Perspectives". **Hewlett-Packard Journal** 1975, pp 2-4.

-. "Interface Standard 488 in Action: Concepts and Capabilities". Presented at Electro 76, Boston, 1976 Electro Professional Program, Electronic Conventions, El Segundo, CA 90245.

-. "Digital Bus Simplifies Instrument System Communication". **EDN**, September 1972.

-. "A Common Digital Interface for Programmable Instruments: The Evolution of a System". **Hewlett-Packard Journal**, October 1972.

-. "Instrument Communications: A New Interface System". INTERCON 1973.

-. "What Makes a Good Interface?". **IEEE Spectrum**, November 1974, pp52-57.

-. "Interface Standard Simplifies Instrumentation Systems". **Canadian Electronics Engineering**, November 1975.

-. "A New INstrument Interface: Needs and Progress Toward a Standard". **ISA Transactions** 14 (1975): 225-30.

Loughry, D.C., and Allen, M.S. "IEEE Standard 488 and Microprocessor Synergism". **Proceedings of the IEEE**, Vol. 66, No. 2, February 1978, pp 162-72.

Madan, P., and Frederick, J. (Intel Corp., Santa Clara, Calif.). "LSI Transceiver Chips Complete GPIB Interface". **Computer Design**, March 1982, pp 91-102.

Marshall, M. "Intelligent Instruments-They Expand Measurement Capabilities, Yet Simplify the Task". **EDN**, April 20, 1977, pp 54-65.

-. "Instrument Makers Score Unannounced Bugs in GPIB Chips". **Electronics**, January 31, 1980, pp39-40.

-. "Intel, TI Upgrade their Chips". **Electronics**, January 31, 1980, pp 44-46.

Mennie, D. (Instrument Editor). "Instrument Interface Systems Directs Test Signals on GPIB". **Electronic Design**, Vol. 27, No. 12, June 7, 1979.

Mihalik, M. (Tektronix, Beaverton, OR). "Microprocessor-Based Product Design Starts with Microprocessor Selection". **Electronic Design**, September 1, 1980, pp 119-25.

Minns, H.G. (Action Micro Systems, Inc., Lake Elmo, MN). "Structured Messages for Control of IEEE-488 Devices-A Proposed Standard", 1980 Electro Professional Program, Electronic Conventions, El Segundo, CA 90245.

Montgomery, Steve and Dash, Mike (Summation, Inc., Woodinville, WA). "Computer-Integrated Testing Meets the Just-In-Time Challenge". **Electronics Test**, October, 1985, pp 36-44.

Motorola (Motorola Semiconductor Products Inc., Phoenix, AZ). "Getting Aboard the 488-1975 Bus". **Publication 11338**, October 1977.

Muche, C.E., and Lin, C.S. "IEEE 488 Applied to Radar and Scientific Systems". Presented at Electro 76 Conference, Session 29, May 1976.

Myles, R. "IEC Interface: A Microprocessor Based Implementation". **IMEKO VII Congress**, London, England. Vol. 3, May 1976.

Nadig, H.J. "A Quiet, HP-IB Compatable Printer that Listens to Both ASCII and BCD". **Hewlett-Packard Journal**, January 1975, pp 14-16.

Nelson, G.E., and Ricci, D.W. "A Practical Interface System for Electronic Instruments" **Hewlett-Packard Journal**, October 1972.

Nelson, J.E. (Hewlett-Packard Co.). "Trends in Instrumentation". Electro 76 Professional Program, Boston, May 11-14, 1976.

-. "Exploding Technology is Shaping the Course of Tomorrow's Instrumentation". **Instruments and Control Systems**, June 1976, pp 43-46.

-. "Trends in Instrumentation". Presented at WESCON 76 Conference, Session 24, 1976.

NIM Committee, National Bureau of Standards, 1972. CAMAC, **A Modular Instrumejntation System for Data handling**. United States Atomic Energy Commission document number TID-25875.

O'Loughlin, Joseph (Dylon corp.,San Diego, CA). "Integrating the IEEE-488 Bus". **Mini-Micro Systems**, June 1980, pp 125-26.

Osborne, A., and Donahue,C.S. **PET/CBM Personal Computer Guide**. 2nd ed. Berkely: OSBORNE/McGraw-Hill, 1980.

Pannach, A. "Interface Entwicklung fur den IEC-Bus". **Elektronik**, 1975, pp 61-64.

Peterson, Steve, "Friendly Software for Test and Measurement", **Computer Design**, October 1983.

Pieper, J., and Grossi, R.J. "LSI Streamlines Instrument Interface with Standard IEEE 488 Bus". **Electronics**, April, 26, 1979, pp 145-50.

Pine, K. "Troubleshooting with Logic Analyzers". **Digital Design**, October 1977, pp 64-75.

Racal-Dana (Racal-Dana Instruments, Inc., Irvine, CA). "Racal-Dana GPIB Test Systems Handbook". **Publication No. 980593.**

Ramirez, Bob and Tilden, Mark (Tektronix Inc., Beaverton, OR). "Personal Computer Analyzes ATE Measurements". **EDN**, February, 21, 1985.

Ramirez, Bob. "Some Useful Approaches to Pulse Analysis", **Handshake**, Vol. 4, N0. 3 (Tektronix newsletter of signal processing and instrument control).

-. "Writing Signal Processing Routines for the 4041 System Controller", **Handshake**, Vol. 6, No. 3.

Randal, William C., and Kerth, Norm. "Microprocessors in instrumentation". **Proceedings of the IEEE**, Vol. 66, No. 2, February 1978.

Rhine, G. "IEEE 488 Applied to a Graphic Terminal", presented at ELECTRO 76 Conference, Session 29, 1976.

Ricci, D.W. (Hewlett-Packard). "IEEE 488: Its Impact on the Design, Building, and Programming of Automatic Test and Measurement Systems". 1980 Electro Professional Program, Electronic Conventions, El Segundo, CA 90245.

Ricci, D.W., and Nelson, G.E. "Standard Instrument Interface Simplifies System Design". **Electronics**, November, 14, 1974, pp 95-106.

Ricci, D.W., and Stone, P.S. "Putting Together Instrument Systems at Minimum Cost". **Hewlett-Packard Journal**, January, 1975, pp 5-11.

Richert, U. "IEC-Bus Interface fur Prozebrechner". **Electronik** 12 (1976): 5-62.

Richter, M. "Das Zustandsdiagramm und seine Anwendung beim IEC-Bus". **Electronik** February, 1977, pp 55-58.

Rony, Peter R. (Virginia Polytechnic Institute and State University, Blacksburg, VA). "Interfacing Fundamentals: Bused Flags". **Computer Design**, July, 1981, pp 162-68.

-. "Interfacing Fundamentals: Conditional I/O Using a Semiphore". **Computer Design**, April, 1980, pp 166-67.

-. "Interfacing Fundamentals: Bidirectional I/O Using Two Semiphores". **Computer Design**, April, 1981, pp 184-88.

"Rs-449 Ousts Rs-232". **Digital Design**, December, 1977, p 18.

Samsen, G.R., and Hudson, R.D. "Bus Adapter Simplifies Interprocessor Communication". **Computer Design**, December 1980, pp 119-24.

Sangren, K.(Sperry Univac, St. Paul, MN). "Instrument Selection and Integration". 1980 Electro Professional Program, Electronic Conventions, El Segundo, CA 90245.

Santoni, A. (EDN Western Editor). "What's Wrong with 488? Not Much, But...". **EDN**, November 22, 1977, pp 48-51.

-. "IEEE-488 Bus Controllers Adapt to a World of Processors". **EDN**, November 5, 1980, pp 69-73.

-. "IEEE 488 Instruments". **EDN**, October 28, 1981, pp 77-94.

-. "IEEE 488 Compatable Instruments". **EDN**, November 5, 1979, pp 91-98.

Santora, Louis F., Jr. (Bell Laboratories, Denver, CO). "IEEE Error Handling Techniques: Pros and Cons". **Computer Design**, June 1981, pp 143-47.
Schultz, S.E., and Trimble, C:R. "Filling in the Gaps-Modular Accessories for Instrument Systems". **Hewlett-Packard Journal**, January 1975, pp 12-13.

Sideris, G. "Instruments '80: Compatable Instrument Clusters Promise Speedier, Smarter Bench Measurements". **Electronic Design**, October 11, 1979, pp 84-98.

Silvernale, L.P. (Wavetek, San Diego, CA). "GPIB Command Structures". 1976 WESCON Professional Program, Los Angeles, September 14-17.

Small, C. "GPIB-Compatable Test Equipment". **Electronic Engineering Times**, November 10,1980, pp53-73.

Smolin, M.; Graves, D.; Winter, K.; and Schwartz, M. "Unified Buses Make the Peripheral IC/OC Connection". **Digital Design**,May 1978. pp 34-44.

-. "Unified Busses Make the Peripheral IC/OC Connection". **Digital Design**, July 1978, pp 86-96.

Socolovsky, A. "The Happy State of a Standard". **EE/Systems Engineering Today", May 1973.**

Staff Report (Benwill Publishing Co.). "Test Instrument Perspective: 'Intelligence' in Test Instruments". Electronics Test, January 1981, p 56.

Stein, P. "Small World-The 488 Interface Interface Bus". **Computer Decisions**, July 1977, p 16.

Sugerman, R. "IEEE Interface Beats Microprocessor as Most -Wanted Sigmal Generator Feature". **Electronic Engineering Times**, February 1976.

Tektronix, Inc. "Tektronix Codes and Formats for GPIB Instruments", **Tektronix Publication 99AX-4607**, 1981

Texas Instruments Incorporated, Engineering Staff, Semiconductor Group. **TMS9914 GPIB Adapter Preliminary Data Manual**, 1979.

Tilden, Mark (tektronix, Inc). "Programming Techniques Speed IEE-488 System Execution". **EDN**, October 27, 1983, pp

–. "Semiconductor Switch Testing, Parts I and II", **Handshake**, Vol. 6, No 3.

Tilden, Mark and Ramirez, Bob (Tektronix, Inc). "Understanding IEEE-488 Basics Simplifies System Integration". **EDN**, June 9, 1982, pp

–. "Gpib Software Configeration Determines System Performance". **EDN**, October 27, 1983, pp

–. "The IBM PC Handles ATE Applications". **EDN**, February 7, 1985, pp

Trifari, J. "Bus Standard Brings New Power to Bench-top Instrumentation", **Electronic Products**, July 1976, pp31-34.

–. "Micros Boost Test Gear Power". **Electronic Products**, October 1976, pp35-39.

Truchard, James J.; Nowlin, William C.; and Kodosky, Jeffery L. (National Instruments Corporation, Austin, TX). "Course 81/63, Microprocessors-10, IEEE-488 and Camac". **Measurement & Data Corporation**, 1980, pp C1-C16.

Turner, Richard L., and Neudorfer, Paul (Dept. of Electrical Engineering, **Seattle** University, Seattle, WA). "Automated Testing - A Course Based on IEEE-488 Equipment of the John Fluke Manufacturing Company". **John Fluke Mfg. Co. Publication**, 1894.

Twaddell, William (Western Editor EDN). "Low-Cost, High-Capacity LSI Chips ease Interface Bus Implementation". **EDN**, November 5, 1979, pp 36-40.

Urdaneta, N.; Chui, C.; and Robinson, J. (Racal-Dana Instruments Inc.). "Structured Programming Cuts Software Costs for ATE". **Electronic Design**, February 4, 1982, pp 107-14.

Urdaneta, Nelson, and King, Bruce (Racal-Dana, Inc., Irvine, CA 92713). "Debug ATE Systems Rapidly with a Low-Cost Bus Analyzer". **Electronic Design**, June 25, 1981, pp 111-115.

Wiewald, J., and West, B. (E-H Research Laboratories Inc., Oakland, CA). "Recent Advances in Microprocessor-Based Test and Measurement Equipment". Electro 76 Professional Program, May 11-14, 1976.

Williams, R. M. "LSI Chips Ease Standard IEEE 488 Bus Interfacing". **Computer Design**, October 1979, pp 123-31.

Wiseman, C.D. "Evaluating GPIB Systems: Bus Analyzers do More Than Test". **Electronics Test**, November 1979, p 26.

-. "Getting on the Bus". **Electronics Test**, June 1979, pp 66-69.

-. "IEEE 488-The General Purpose Interface Bus: Pitfalls and Prospects". **Electronics Test**, April 1981, pp 52-63.

Wolpert, D.L. "Multifunction Scanner for Calculator based Data Acquisition Systems". **Hewlett-Packard Journal**, January 1975, p 17.

Yencharis, L. "Expansion of IEEE Interface to Test Equipment Cited as Main Improvements: Microprocessor still not a Major Addition". **Elexctronic Engineering Times**, June 27, 1977, pp 16-19.

Young, R. "Implementing an IEEE-488 Bus Controller with Microprocessor Software". **IEEE Transactions On IECI. IECI-27**, February 1980, pp10-15.

APPENDIX C

RECOMMENDED EQUIPMENT LIST

1. CONTROLLER

A. HP 85B & 87 - These controllers, although slow for many industrial applications, work well in the educational lab environment. They are very user friendly, and their manuals are well written. The HP 87 has been replaced by the HP 86. Both support multiple GPIB ports as well as RS-232, and general purpose digital (I/O which can be used to control lab constructed scanner switches, or control non GPIB instruments i.e. bcd, etc.) These and the 87 may be purchased from equipment rental firms, often with full warranty, such as Electrorent, Corp. These are programmed in a HP enhanced BASIC. 1987 price from about $2500 to $3500.

B. Fluke 1722A (and earlier model 1720A) are instrument controllers which feature a touch sensitive screen (keyboard may be detached and programs run from touch screen menus). Programmed in BASIC (or FORTRAN). Rack mountable, so it may be built into a "rack & stack" test station. The 1722A supports multiple GPIB ports, a RS-232 port, as well as two 16 bit parallel I/O ports. For advanced programming a compiled and extended BASIC are available. 1986 pricing $6400.

C. IBM PC (and its clones) - These can be used as instrument controllers with add on cards from National Instruments, Austin, TX (very popular among add on cards); Hewlett Packard, Palo Alto, CA; I/O Tech, Cleveland, OH; and others (see bibliography). Since the PC was not designed as an instrument controller, it is not as fast, nor as easily programmed as the above controllers.

D. Commodore PET & C-64 - The PET has a GPIB port. A special cable PET - GPIB is needed to allow the PET to be used as a controller. The C-64 a serial bus which follows GPIB protocol. Computer Specialities Inc., Melbourne, FLA manufacturers a serial bus - GPIB converter (400S). A good reference on PET (and C-64) as instrument controllers is "PET/CBM and the IEEE Bus (GPIB)", by Fisher and Jensen (see bibliography). Since these were not designed as instrument controllers they are relatively slow and have limited capabilities.

2. SYSTEM VOLTMETERS

A. HP 3478A - This is a 5 1/2 digit electronically calibrated DMM. It features a 12 digit LCD display, which may be programmed to display operator prompts. Features: DC & true RMS AC voltage and current, as well as 2 & 4 wire ohms. With lower resolution it may take up to 90 readings/ second. 1987 price $995.

B. Fluke 8840A - This is a 5 1/2 digit electronically calibrated DMM. It utilizes a 7-segment vacuum flourescent display (not user accessable). Features: DC voltage and current, plus 2 & 4 wire ohms. 1986 prices $760, GPIB interface - $150, true RMS AC - $185.

C. Keithly produces a low cost GPIB DMM, but none of its functions or ranges are programmable, a severe limitation for a total learning experience. It also does not have service request (SRQ) capability.

2. FUNCTION GENERATORS

A. HP 3314A - This is a 1mHz to 20MHz generator.Features: sine, square, triangle waveforms with symmetry, offset, sweep, burst, AM, and FM. It also features the capability for user created (arbitrary) waveforms i.e heart, simulate mechanical vibration, etc. All front panel controls are programmable. 1987 price $4200.

B. HP 8116A - This is a 1mHz to 50MHz pulse/function generator. Functions similar to 3314A, no arbitrary waveform capability. 1987 prices $3500, burst & log sweep option - $400.

C. HP 3325A - This is a 1uHZ to 21MHz function/signal synthesizer, with accuracies beyond the above two instruments. No symmetry may be a limiting factor. 1987 price $4500.

3. SCANNER SWITCHES

A. Fluke 2205A - The basic box will handle 10 cards, with each card able to switch up to 10 points. May be expanded to 100 cards (1000 points). Multiple scanner boxes may be used in a system. 1986 prices $2400, plus $360/card.

B. HP 3488A - The basic box will handle up to 5 cards, with each card able to switch up to 10 points. Multiple boxes may be used in a system. 1987 prices $1400, plus $400/card.

4. COUNTER TIMERS

A. Fluke 1953A - Will count to 125MHz (higher with option). Features: Frequency, frequency ratio, period, period averaging, time interval, gated totalize measurements. Bus control of function, range, and signal conditioning. 1986 price $1750.

B. HP 5316A - Same features as Fluke above. 1987 price $1900.

5. BUS ANALYZERS

1. Racal-Dana 488 - An easy to use, battery powered, bus analyzer will store 40-bytes of bus activity. Output is in hexadecimal. The instrument can be set at the beginning, middle or end of the 40-byte window. The status of other lines, such as ATN, is also given allowing one to differentiate between data and command. Very handy for showing students exactly what is traveling on the bus. Students learn to identify who is talking and who is listening. Although production has been stopped on this instrument, it is available from equipment rental firms, and on the used equipment market. Origional cost $995.

2. Logic Analyzers (from a variety of manufacturers) may be used to moniyor bus activity.

OTHER MANUFACTURERS

Tektronix manufacturers a modular series that reside in a main frame (TM 5000 series). These tend to be costlier than the above instruments.

APPENDIX D

SUGGESTED READING LIST

See bibliography for complete details on reading references.

1. "PET/CBM and the IEEE 488 Bus (GPIB) - This is one of the few books that have been published which deal specifically with the GPIB, a good introduction.

2. "Tutorial Description of the Hewlett-Packard Interface Bus" - The most comprehensive "down to earth" technical description of the GPIB. Free from HP, well worth reading and for use as a class reference.

3. "Understanding IEEE-488 Basics Simplifies System Integration", "GPIB Software Configeration Determines System Performance", "Many Choices Complicate IEEE-488 Instrument Selection" - Well written general information, available in reprint form from Tektronix.

4. "Racal-Dana GPIB Test Systems Handbook" - Group of articles on a variety of GPIB topics, well written and practical. Available from Racal-Dana.

5. "Using the 3478A" - Well written and practical guide to using the 3478A DMM. Available from HP.

6. "IEEE Standard Digital Interface for Programmable Instrumentatiom" - The official definition of the GPIB and its requirements. Difficult to read, but good to have on hand as a reference. Available from the IEEE.

7. "Automated Testing" - Good when using Fluke controller and instrumentation. Good introduction to Fluke instrumentation through lab exercises. Available from Seattle University.

APPENDIX E

EXAM & EXAM SOULTIONS

1. A suggested final exam for this curriculum follows this page.

2. Solutions for suggested final exam:

1. D
2. D
3. B
4. B
5. D
6. E
7. A
8. B
9. C
10. B
11. C
12. E
13. A
14. D
15. C
16. A
17. B
18. D
19. A
20. B
21. B
22. B
23. D
24. A
25. B
26. B
27. A
28. E
29. D
30. A

FINAL EXAM COMPUTER CONTROLLED TEST SYSTEMS

Directions: This exam is divided into two parts, a multiple choice portion and a subjective portion. You have two hours to complete this exam. If you are not complete with the first portion after one hour go on to the second half of the exam. On the objective portion of the exam choose the **BEST SINGLE** answer and place its letter next to the corrosponding number on your answer card. Be sure to place your full name, class number, class section, and date on **BOTH** test sheets. **DO NOT** write on this exam!

1. Which of the following are other names for the IEEE 488 bus?
   A. GPIB
   B. HPIB
   C. ASCII bus
   D. All of the above
   E. A & B only

2. Which of the following is **NOT** defined by the IEEE 488 standard?
   A. Functional aspects
   B. Electrical aspects
   C. Mechanical aspects
   D. Operational aspects
   E. A & D only

3. How many devices **INCLUDING** a controller may be connected to one contiguous bus?
   A. 14
   B. 15
   C. 31
   D. 32
   E. There is no limit to the number of devices which may be connected to one contiguous bus.

4. The IEEE 488 uses ________ logic with _________ polarity in reference to ground.
   A. Positive true, positive
   B. Negative true, positive
   C. Positive true, negative
   D. Negative true, negative
   E. None of the above

5. The IEEE 488 uses a ________ bit ________ data bus.
   A. 7, serial
   B. 8, serial
   C. 7, parallel
   D. 8, parallel
   E. Either A or B

ES-2

6. Which of the following is FALSE regarding the IEEE 488 bus?
 A. Data transfer rates up to 1Megabyte/second is possible.
 B. There may be only one active talker on the bus at a time.
 C. There may be several active listeners on the bus at a time.
 D. Maximum cable length is 20 meters or 2 meters per devive connected to the bus.
 E. None of the above statements are false.

7. All powered devices on the bus (designed to respond) must respond to universal commands.
 A. True
 B. False

8. The speed of the handshake process is determined by the ______ device on the bus.
 A. Fastest
 B. Slowest
 C. Speed is preset, and not determined by devices connected to the bus.

9. To insure proper bus operation ____ of the devices connected to the bus must be powered on (whether they are being used or not).
 A. 1/3
 B. 1/2
 C. 2/3
 D. All

10. When ATN is low (true) the IEEE 488 is in which transfer mode?
 A. Command (can send addresses only)
 B. Command (can send commands and addresses only)
 C. Command (can send commands only)
 D. Data (can send device dependent data only)
 E. None of the above

11. Sending a BASIC command which incorporates the IEEE 488 command DCL (device clear) will cause __________________. NOTE: All devices designed to respond to DCL.
 A. the device addressed to talk to go to its default state
 B. devices addressed to listen to go to their default state
 C. all devices will go to their default state
 D. None of the above

12. Which is the preferred method for interconnecting IEEE 488 instruments?
 A. Star
 B. Linear
 C. Star-linear
 D. Linear-star
 E. What ever is most convienent for the particular test setup being used.

13. Which of the following is not a universal command?
 A. GET
 B. LLO
 C. UNT
 D. UNL
 E. All of the above are universal commands.

14. Which of the following bus management lines is used to indicate message termination (often in place of CrLf)?
- A. IFC
- B. REN
- C. ATN
- D. EOI
- E. None of the above can serve in that capacity.

15. How many different primary addresses can be used to select instruments (one is not useable in this capacity)?
- A. 14
- B. 15
- C. 31
- D. 32
- E. 64

16. Must the controller have an address?
- A. Yes
- B. No
- C. Only if two, or more, controllers are on the same bus.

17. If you want two DMMs to take the same readings can they have the same address?
- A. Yes
- B. No

18. A bus analyzer records the hex code 28, the device is at address ______ (decimal) and is being setup as a __________. ATN=true
- A. 40, listener
- B. 40, talker
- C. 8, talker
- D. 8, listener
- E. 28, listener

19. In HP BASIC, the command ENTER 723; v$ is executed. What is the hex code as seen on a bus analyzer which sets the controller address (assume controller address is 21 decimal)?
- A. 35
- B. 55
- C. 37
- D. 57
- E. None of the above

Questions 20-24 refer to the following information. The following hex sequence is recorded on a bus analyzer. Data is in ASCII.

ATN	T	T	T	F	F	F	F	F	F	F	F	F	F	F
	3F	38	52	2B	33	2E	30	39	32	45	2D	36	0D	0A
BYTE	1	2	3	4	5	6	7	8	9	10	11	12	13	14

TIME-----------)

Assume controller is addressed first, then the DMM.

20. Information is being sent _____.
A. from the controller to the DMM, to set it up
B. from the DMM to the controller i.e. a reading

21. How many commands bytes (as opposed to data bytes) are being sent across the bus?
A. 1
B. 3
C. 9
D. 11
E. 14

22. What is being used to terminate the message?
A. No termination is being used.
B. CrLf
C. A bus management line provides termination.
D. None of the above

23. What data message is being sent across the bus?
A. Setup: autorange, DCV, single trigger
B. Setup: range 3, ACV, external trigger
C. Serial poll status byte
D. +3.092E-6CrLf
E. 43,51,46,48,57,50,69,45,54,13,10

24. In reference to the concepts of structured programming, modules should be _______ bound.
A. tightly
B. loosly

25. In reference to the concepts of structured programming, modules should be _______ coupled.
A. tightly
B. loosly

26. In writing structured modules it is best to limit the number of varables to ______.
A. 1
B. 7
C. 13
D. 40-50
E. No limit is set regarding the number of variables in a module.

27. If a test system failes to function on its first run, which of the following is **least** likely to be the cause of the failure?
A. Hardware malfunction
B. Coding error
C. Data format incompatability

28. Which of the following are TRUE in reference to using personal computers as instrument controllers?
   - A. Command languages are very cryptic and difficult for beginners to use.
   - B. Real time SRQs (service requests) are not available.
   - C. A minimal number of seperate ports are available.
   - D. They are very inexpensive, as compared to computers specifically designed as instrument controllers.
   - E. All of the Above are true

29. Which of the following are TRUE in reference to bus extenders?
   - A. They can be used to extend cable lengths beyond the IEEE 488 specified maximum.
   - B. They can be used to extend the number of devices on one bus beyond the specified maximum.
   - C. They are transparant i.e. they do not need to be addressed.
   - D. All of the above are true
   - E. Only A & C are true

30. Which of the following methods is the quickest way to take 100 readings and average them?
   - A. Use an intelligent instrument i.e. tell it to take 100 readings, average them, and send the result to the controller.
   - B. Have each reading sent to the controller, which then averages them.

ES-6

31. Part II. On a seperate piece of paper construct a structure chart which will describe a system that has the following characteristics and attributes.

Project title: Incoming inspection IC tester

Mission statement: To develop the software capable of all necessary tasks related to testing incomming ICs.

Objectives:

1. The program should be designed to be compatable with three different types of DMMs and function generators, and two types of controllers.
2. The program should be designed to test 10 different digital ICs on one test fixture, and 5 different analog ICs on a different test fixture.
3. Programmable scanner switches are used to switch instruments to different test points in the test fixture.
4. The test fixtures are identified by the system by reading a fixed value resistor at preset test points.
5. Each ICs go-no go test limit data is stored in an ASCII file on disk.
6. Each ICs test fixture setup data is stored in a ASCII file on a disk.
7. The instruments in a particular "rack and stack" must be identified by software, and tested to see if they function.
8. The user interface should be "friendly" and simple.
9. The following data items should be logged:part #, test results - go-no go, operator, vendor, date.
10. The system should reboot and resumption of testing should occur on resumption of power.

ES-7